



TRABALHO DE CONCLUSÃO DE CURSO

**Aplicação de um modelo e método de extração unificado (UMEM)
para circuitos com transistores verticais de filmes finos orgânicos**

Lucas Monteiro Torres

Brasília, Julho de 2016

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

TRABALHO DE CONCLUSÃO DE CURSO

**Aplicação de um modelo e método de extração unificado (UMEM)
para circuitos com transistores verticais de filmes finos orgânicos**

Lucas Monteiro Torres

*Relatório submetido ao Departamento de Engenharia
Elétrica como requisito parcial para obtenção
do grau de Engenheiro Eletricista*

Banca Examinadora

Prof. Stefan Blawid, ENE/UnB
Orientador

Prof. José Camargo Costa, ENE/UnB
Examinador interno

Prof. Daniel Chaves Café, ENE/UnB
Examinador interno

RESUMO

Para este projeto, inicialmente, a implementação do UMEM é transferida do MATHCAD para o MATLAB, uma plataforma mais conhecida e difundida. Em seguida, o código é aprimorado, com o propósito de otimizar seu uso e torná-lo mais genérico para OTFTs em geral. Essas modificações incluem o desenvolvimento de um algoritmo a partir de algumas recomendações fornecidas pelo código no MATHCAD, assim como a otimização dos parâmetros ajustáveis do modelo. Com a adoção de duas formas de obtê-los, algumas observações são feitas em relação às limitações impostas pelo número de medidas necessárias das curvas experimentais. Na próxima seção do projeto, a implementação do modelo em Verilog A (QUCS) é modificada, a fim de compatibilizar o uso do UMEM nas duas plataformas (MATLAB e QUCS) e usá-lo em um simulador de circuitos. Em seguida, um estudo de caso é realizado (aplicação do UMEM à um VOTFT), para o qual o dispositivo é simulado em quatro topologias de amplificadores inversores. Nesse estágio do projeto, é observado que o número de medidas das curvas experimentais afeta as características estática de transferência. Por fim, algumas recomendações são feitas em relação à obtenção das curvas experimentais, aplicação do UMEM à um transistor tipo N, estudo de casos em AC e associação do método de extração do UMEM à outros modelos da literatura.

ABSTRACT

In this project, initially, the UMEM implementation is transferred from MATHCAD to MATLAB, a well known and widespread platform. Hence, the code is improved, in order to optimize its usage and make it applicable for OTFTs in general. These changes include the development of an algorithm based on some recommendations provided by the code in MATHCAD, as well as an optimization of the model adjustable parameters. Along with the adoption of two methods, some observations are made regarding limitations imposed by the number of measurements from experimental curves. In the project following section, the model implementation in Verilog A (QUCS) is modified in order to make both platforms (MATLAB and QUCS) achieve the same results (output and transfer curves), therefore, extracted parameters can be used in circuit simulations. Then, a case study is performed (UMEM applied to a VOTFT), for which the device is simulated in four inverting amplifier topologies. In that stage, it is observed that the number of measurements from experimental curves also affects voltage transfer characteristics. Finally, some recommendations are made with respect to experimental curves, extraction of N type transistor parameters, AC case studies and addition of UMEM extraction method to other models in the literature.

SUMÁRIO

0.1	ABREVIACÕES.....	11
0.2	PARÂMETROS FÍSICOS DOS TRANSISTORES.....	11
0.3	SÍMBOLOS DOS PARÂMETROS/FUNÇÕES ASSOCIADOS À EXTRAÇÃO DO UMEM	12
0.4	SÍMBOLOS DOS PARÂMETROS/FUNÇÕES ASSOCIADOS AO MODELO UMEM	13
1	Introdução.....	1
1.1	CONTEXTUALIZAÇÃO.....	1
1.2	DEFINIÇÃO DO PROBLEMA	1
1.3	OBJETIVOS DO PROJETO	2
1.4	APRESENTAÇÃO DO RELATÓRIO	2
2	Revisão Bibliográfica.....	3
2.1	INTRODUÇÃO.....	3
2.2	TRANSISTORES VERTICAIS DE FILMES FINOS ORGÂNICOS (VOTFT).....	3
2.2.1	TRANSISTORES ORGÂNICOS DE BASE PERMEÁVEL (OPBT)	4
2.2.2	TRANSISTORES ORGÂNICOS DE INDUÇÃO ESTATICA (OSIT).....	5
2.2.3	TRANSISTORES ORGÂNICOS DE BARREIRA SCHOTTKY (OSBT).....	5
2.2.4	TRANSISTORES ORGÂNICOS DE EFEITO DE CAMPO (VOFET)	6
2.3	MODELOS COMPACTOS PARA OTFT'S.....	7
2.4	O MODELO E MÉTODO DE EXTRAÇÃO UNIFICADOS (UMEM).....	8
2.4.1	ALGORÍTIMO DE EXTRAÇÃO DO UMEM	9
2.4.2	MODELO UMEM	12
2.5	AMPLIFICADORES INVERSORES	13
2.5.1	AMPLIFICADORES INVERSORES CMOS	15
2.6	VERILOG A E QUCS.....	18
3	Desenvolvimento.....	20
3.1	INTRODUÇÃO.....	20
3.2	SUBSTITUIÇÃO DO CÓDIGO NO MATHCAD POR UM EQUIVALENTE NO MA- TLAB	20
3.3	APRIMORAMENTO DO CÓDIGO NO MATLAB	24
3.3.1	OTIMIZAÇÃO DOS PARÂMETROS DE AJUSTE DA EXTRAÇÃO	26
3.3.2	OTIMIZAÇÃO DOS PARÂMETROS AJUSTÁVEIS DO MODELO	28
3.4	MODIFICAÇÕES DO CÓDIGO EM VERILOG A	29

3.5	APLICAÇÕES DE VOTFT EM TOPOLOGIAS DE CIRCUITO	30
4	Resultados.....	32
4.1	INTRODUÇÃO.....	32
4.2	VALIDAÇÃO DA IMPLEMENTAÇÃO DO UMEM NA PLATAFORMA MATLAB	32
4.3	OTIMIZAÇÃO DOS PARÂMETROS DE AJUSTE DA EXTRAÇÃO DO UMEM NO MATLAB.....	35
4.3.1	OTIMIZAÇÃO DOS PARÂMETROS DE AJUSTE DA EXTRAÇÃO PARA O TUD OTFT	35
4.3.2	OTIMIZAÇÃO DOS PARÂMETROS DE AJUSTE DA EXTRAÇÃO PARA O TUD VOTFT	38
4.4	OTIMIZAÇÃO DOS PARÂMETROS AJUSTÁVEIS DO MODELO UMEM.....	43
4.4.1	OTIMIZAÇÃO DOS PARÂMETROS AJUSTÁVEIS DO MODELO PARA O TUD OTFT	44
4.4.2	OTIMIZAÇÃO DOS PARÂMETROS AJUSTÁVEIS DO MODELO PARA O TUD VOTFT	48
4.5	VALIDAÇÃO DA IMPLEMENTAÇÃO DO MODELO UMEM NA PLATAFORMA QUCS	52
4.6	APLICAÇÕES DO TUD VOTFT EM TOPOLOGIAS DE CIRCUITO USANDO QUCS	56
4.6.1	AMPLIFICADOR FONTE COMUM	56
4.6.2	AMPLIFICADOR INVERSOR COM CARGA ATIVA	56
4.6.3	AMPLIFICADOR INVERSOR COM FONTE DE CORRENTE.....	57
4.6.4	AMPLIFICADOR INVERSOR PUSH-PULL.....	59
5	Conclusões.....	62
5.1	RECOMENDAÇÕES PARA TRABALHOS FUTUROS	63
	REFERÊNCIAS BIBLIOGRÁFICAS	65
	Anexos.....	66
I	Código no MATLAB.....	67
I.1	MÓDULO DA DEFINIÇÃO DOS PARÂMETROS FÍSICOS.....	67
I.2	MÓDULO DO CARREGAMENTO E ARMAZENAMENTO DOS DADOS CORRESPON- DENTES ÀS CURVAS EXPERIMENTAIS	68
I.3	MÓDULO COM AS INTERPOLAÇÕES	69
I.4	MÓDULO COM A IMPLEMENTAÇÃO DO UMEM (EXTRAÇÃO + MODELO).....	73
II	Código em Verilog A.....	87
III	Anexos da seção 4.3.....	91
IV	Anexos da seção 4.4.....	97

LISTA DE FIGURAS

2.1	Arquitetura de um transistor vertical vista a partir de uma das arquiteturas usadas para OTFT's. O fluxo de cargas ocorre a partir dos dois eletrodos correspondentes à fonte em direção ao dreno, verticalmente.	4
2.2	Diagrama correspondente ao princípio de funcionamento de um OPBT.	4
2.3	Arquitetura correspondente ao OPBT.	5
2.4	Procedimentos para construção de um OSIT.	6
2.5	Arquitetura correspondente ao OSBT.	6
2.6	Arquitetura correspondente ao VOFET usado neste projeto.	6
2.7	Diferentes arquiteturas para um OTFT. Suas denominações são feitas de acordo com a posição relativa dos contatos em relação a camada do semiconductor. Nas estruturas coplanares, dreno e fonte estão no mesmo plano com o canal, enquanto que nas estruturas escalonadas, o canal está separado dos eletrodos por extensões do semiconductor. Para estruturas escalonadas, o aumento da área de contato dos eletrodos reduz a resistência na injeção de portadores.	8
2.8	Topologia básica correspondente à um amplificador inversor fonte comum.	14
2.9	Curva de transferência de tensão de um amplificador inversor fonte comum.	14
2.10	Curva de transferência de tensão de um amplificador inversor com carga ativa.	15
2.11	Modelo de pequenos sinais para um amplificador inversor com carga ativa.	16
2.12	Curva de transferência de tensão de um amplificador inversor com fonte de corrente.	17
2.13	Curva de transferência de tensão de um amplificador inversor Push-pull.	18
2.14	Modelo de pequenos sinais para um amplificador inversor Push-pull.	18
3.1	Fluxograma do projeto. Blocos em azul contém o fluxo para simulações de quaisquer OTFTs. Blocos em verde contém etapas realizadas somente para este projeto. A realimentação é feita apenas uma vez. A constante n indica um número suficiente de vezes que o bloco deve ser repetido antes do caminho direto ser executado novamente.	21
3.2	Trecho do código no MATLAB antes (parte comentada) e após alteração da forma de calcular os parâmetros da reta $HH1(V_{gs})$	22
3.3	Trecho do código no MATLAB antes (parte comentada) e após alteração da forma de calcular o coeficiente angular da reta $IPC(V_{gs})$	23
3.4	Trecho do código no MATLAB mostrando como V_T é localizado no vetor corrente I_{ds} com a consideração de três casas decimais (multiplicação de V_T por 1000).	23

3.5	Trecho do código no MATLAB mostrando como $H(V_{gs})$ é determinado, através da localização de pontos em um vetor com valores das integrais calculadas para todas as medidas de corrente. O número comentado mostra qual era o tamanho do vetor após interpolação, informação que não precisa mais ser inserida.	25
3.6	Trecho do código no MATLAB mostrando como a curva de saída apropriada, <i>imaxcor</i> , é escolhida para determinação dos parâmetros λ e m	25
3.7	Interface do QUCS com o usuário para inserção de parâmetros do modelo em cada transistor.	31
4.1	Curvas características de saída do CINVESTAV OTFT encontradas experimentalmente e modeladas pelo UMEM no MATLAB e MATHCAD, usando os parâmetros da tabela 4.1 para $V_{gs} = 0V, -10V, -20V$ e $-30V$	33
4.2	Curvas características de transferência do CINVESTAV OTFT encontradas experimentalmente e modeladas pelo UMEM no MATLAB e MATHCAD, usando os parâmetros da tabela 4.1	33
4.3	Curvas características de transferência do CINVESTAV OTFT encontradas experimentalmente e modeladas pelo UMEM no MATLAB e MATHCAD, usando os parâmetros da tabela 4.1	34
4.4	Curvas características de saída para o TUD OTFT encontradas experimentalmente e modelada pelo MATLAB usando os parâmetros da primeira simulação na tabela 4.2 para $V_{gs} = 0V, -2V, -4V$ e $-6V$. $V_{H,min} = -4V$, $V_{H,max} = -6V$, $V_{IPC,min} = -4V$, $V_{IPC,max} = -6V$	35
4.5	Curvas características de saída para o TUD OTFT encontradas experimentalmente e modelada pelo MATLAB usando os parâmetros da segunda simulação na tabela 4.2 com $V_{gs} = 0V, -2V, -4V$ e $-6V$. $V_{H,min} = -3V$, $V_{H,max} = -5V$, $V_{IPC,min} = -4V$, $V_{IPC,max} = -6V$	36
4.6	Curvas características de saída para o TUD OTFT encontradas experimentalmente e modelada pelo MATLAB usando os parâmetros da terceira simulação na tabela 4.2 com $V_{gs} = 0V, -2V, -4V$ e $-6V$. $V_{H,min} = -2V$, $V_{H,max} = -4V$, $V_{IPC,min} = -4V$, $V_{IPC,max} = -6V$	36
4.7	Curvas das funções de ajuste $H(V_{gs})$ e $HH1(V_{gs})$ para o TUD OTFT usando os parâmetros de ajuste da primeira simulação na tabela 4.2. $V_{H,min} = -4V$, $V_{H,max} = -6V$, $V_{IPC,min} = -4V$, $V_{IPC,max} = -6V$	38
4.8	Curvas das funções de ajuste $IPC(V_{gs})$ e $IIS1(V_{gs})$ para o TUD OTFT usando os parâmetros de ajuste da primeira simulação na tabela 4.2. $V_{H,min} = -4V$, $V_{H,max} = -6V$, $V_{IPC,min} = -4V$, $V_{IPC,max} = -6V$	39
4.9	Curvas das funções de ajuste $H(V_{gs})$ e $HH1(V_{gs})$ para o TUD VOTFT usando os parâmetros de ajuste da primeira simulação na tabela 4.4. $V_{H,min} = -4V$, $V_{H,max} = -6V$, $V_{IPC,min} = -3V$, $V_{IPC,max} = -5V$	40
4.10	Curvas das funções de ajuste $H(V_{gs})$ e $HH1(V_{gs})$ para o TUD VOTFT usando os parâmetros de ajuste da terceira simulação na tabela 4.4. $V_{H,min} = -3.5V$, $V_{H,max} = -5.5V$, $V_{IPC,min} = -1.8V$, $V_{IPC,max} = -2.2V$	40

4.11	Curvas características de transferência para o TUD VOTFT encontradas experimentalmente e modelada pelo MATLAB usando os parâmetros da segunda simulação na tabela 4.4. $V_{H,min} = -4V$, $V_{H,max} = -6V$, $V_{IPC,min} = -1.8V$, $V_{IPC,max} = -2.2V$.	41
4.12	Curvas características de transferência para o TUD VOTFT encontradas experimentalmente e modelada pelo MATLAB usando os parâmetros da terceira simulação na tabela 4.4. $V_{H,min} = -3.5V$, $V_{H,max} = -5.5V$, $V_{IPC,min} = -1.8V$, $V_{IPC,max} = -2.2V$.	42
4.13	Curvas características de transferência para o TUD VOTFT encontradas experimentalmente e modelada pelo MATLAB usando os parâmetros da segunda simulação na tabela 4.4. $V_{H,min} = -4V$, $V_{H,max} = -6V$, $V_{IPC,min} = -1.8V$, $V_{IPC,max} = -2.2V$.	42
4.14	Curvas características de transferência para o TUD VOTFT encontradas experimentalmente e modelada pelo MATLAB usando os parâmetros da terceira simulação na tabela 4.4. $V_{H,min} = -3.5V$, $V_{H,max} = -5.5V$, $V_{IPC,min} = -1.8V$, $V_{IPC,max} = -2.2V$.	43
4.15	Curvas características de saída para o TUD VOTFT encontradas experimentalmente e modelada pelo MATLAB usando os parâmetros da segunda simulação na tabela 4.4 com $V_{gs} = 0V, -2V, -4V$ e $-6V$. $V_{H,min} = -4V$, $V_{H,max} = -6V$, $V_{IPC,min} = -1.8V$, $V_{IPC,max} = -2.2V$.	43
4.16	Curvas características de saída para o TUD VOTFT encontradas experimentalmente e modelada pelo MATLAB usando os parâmetros da terceira simulação na tabela 4.4 com $V_{gs} = 0V, -2V, -4V$ e $-6V$. $V_{H,min} = -3.5V$, $V_{H,max} = -5.5V$, $V_{IPC,min} = -1.8V$, $V_{IPC,max} = -2.2V$.	44
4.17	Curvas das funções de ajuste $IPC(V_{gs})$ e $IIS1(V_{gs})$ para o TUD VOTFT usando os parâmetros de ajuste da primeira simulação na tabela 4.4. $V_{H,min} = -4V$, $V_{H,max} = -6V$, $V_{IPC,min} = -3V$, $V_{IPC,max} = -5V$.	44
4.18	Curvas características de transferência para o TUD VOTFT encontradas experimentalmente e modelada pelo MATLAB usando os parâmetros da primeira simulação na tabela 4.4. $V_{H,min} = -4V$, $V_{H,max} = -6V$, $V_{IPC,min} = -3V$, $V_{IPC,max} = -5V$.	45
4.19	Curvas características de transferência para o TUD VOTFT encontradas experimentalmente e modelada pelo MATLAB usando os parâmetros da primeira simulação na tabela 4.4. $V_{H,min} = -4V$, $V_{H,max} = -6V$, $V_{IPC,min} = -3V$, $V_{IPC,max} = -5V$.	45
4.20	Curvas características de saída para o TUD VOTFT encontradas experimentalmente e modelada pelo MATLAB usando os parâmetros da primeira simulação na tabela 4.4 com $V_{gs} = 0V, -2V, -4V$ e $-6V$. $V_{H,min} = -4V$, $V_{H,max} = -6V$, $V_{IPC,min} = -3V$, $V_{IPC,max} = -5V$.	46
4.21	Curvas características de saída para o TUD OTFT encontradas experimentalmente e modelada pelo MATLAB usando os parâmetros ajustáveis do modelo na tabela 4.5 com $V_{gs} = 0V, -2V, -4V$ e $-6V$.	46

4.22	Curvas características de transferência para o TUD OTFT encontradas experimentalmente e modeladas pelo MATLAB usando os parâmetros ajustáveis do modelo na tabela 4.5. Nos gráficos, $V_{gs,SAT} = -1.95V$, $V_T = -1.1932V$. Os pontos correspondentes às medições estão em vermelho. Os demais, em azul, são resultados da interpolação.	47
4.23	Curvas características de transferência para o TUD OTFT encontradas experimentalmente e modeladas pelo MATLAB usando os parâmetros ajustáveis do modelo na tabela 4.5. Nos gráficos, $V_{gs,SAT} = -10.31V$, $V_T = -1.1932V$	47
4.24	Curvas de transcondutância para o TUD OTFT encontradas experimentalmente e modeladas pelo MATLAB usando os parâmetros ajustáveis do modelo na tabela 4.5. Nos gráficos, $V_{gs,SAT} = -10.31V$, $V_T = -1.1932V$	47
4.25	Curvas de transcondutâncias para o TUD OTFT encontradas experimentalmente e modeladas pelo MATLAB usando os parâmetros ajustáveis do modelo na tabela 4.5. Nos gráficos, $V_{gs,SAT} = -1.95V$, $V_T = -1.1932V$. A curva de transcondutância experimental é obtida a partir da derivada da curva interpolada de transferência. Os pontos em vermelho indicam as tensões nas quais a corrente foi medida.	48
4.26	Curvas características de transferência para o TUD VOTFT encontradas experimentalmente e modeladas pelo MATLAB usando os parâmetros ajustáveis do modelo na tabela 4.6. Nos gráficos, $V_{gs,SAT} = -1.13V$, $V_T = -0.676V$. Os pontos correspondentes às medições estão em vermelho. Os demais, em azul, são resultados da interpolação.	49
4.27	Curvas características de transferência para o TUD VOTFT encontradas experimentalmente e modeladas pelo MATLAB usando os parâmetros ajustáveis do modelo na tabela 4.6. Nos gráficos, $V_{gs,SAT} = -5.23V$, $V_T = -0.676V$	49
4.28	Curvas de transcondutância para o TUD VOTFT encontradas experimentalmente e modeladas pelo MATLAB usando os parâmetros ajustáveis do modelo na tabela 4.6. Nos gráficos, $V_{gs,SAT} = -1.13V$, $V_T = -0.676V$. A curva de transcondutância experimental é obtida a partir da derivada da curva interpolada de transferência. Os pontos em vermelho indicam as tensões nas quais a corrente foi medida.	50
4.29	Curvas de transcondutâncias para o TUD VOTFT encontradas experimentalmente e modeladas pelo MATLAB usando os parâmetros ajustáveis do modelo na tabela 4.6. Nos gráficos, $V_{gs,SAT} = -5.23V$, $V_T = -0.676V$	51
4.30	Curvas características de saída encontradas experimentalmente e modeladas pelo MATLAB e QUCS para $V_{gs} = 0V$, $-10V$, $-20V$ e $-30V$	53
4.31	Curvas características de transferência para o CINESTAV OTFT encontradas experimentalmente e modeladas pelo MATLAB e QUCS para $V_{ds} = -30V$	53
4.32	Curvas características de transferência para o CINESTAV OTFT encontradas experimentalmente e modeladas pelo MATLAB e QUCS para $V_{ds} = -5V$	54
4.33	Curvas características de transferência para o TUD OTFT encontradas experimentalmente e modeladas pelo MATLAB e QUCS para $V_{ds} = -0.5V$	54
4.34	Curvas características de transferência encontradas experimentalmente e modeladas pelo MATLAB e QUCS.	55

4.35	Curvas de transcondutâncias para o TUD OTFT encontradas experimentalmente e modeladas pelo MATLAB e QUCS para $V_{ds} = -0.5V$.	55
4.36	Amplificador fonte comum.	57
4.37	Característica estática de transferência de tensão do amplificador fonte comum.	57
4.38	Amplificador inversor com carga ativa.	58
4.39	Característica estática de transferência de tensão do amplificador inversor com carga ativa.	58
4.40	Amplificador inversor com fonte de corrente.	59
4.41	Característica estática de transferência de tensão do amplificador inversor com fonte de corrente.	60
4.42	Amplificador inversor push-pull.	60
4.43	Característica estática de transferência de tensão do amplificador inversor push-pull.	61
III.1	Curvas características de transferência para o TUD OTFT encontradas experimentalmente e modelada pelo MATLAB para primeira simulação da tabela 4.2. $V_{H,min} = -4V$, $V_{H,max} = -6V$, $V_{IPC,min} = -4V$, $V_{IPC,max} = -6V$.	91
III.2	Curvas características de transferência para o TUD OTFT encontradas experimentalmente e modelada pelo MATLAB para segunda simulação da tabela 4.2. $V_{H,min} = -3V$, $V_{H,max} = -5V$, $V_{IPC,min} = -4V$, $V_{IPC,max} = -6V$.	91
III.3	Curvas características de transferência para o TUD OTFT encontradas experimentalmente e modelada pelo MATLAB para terceira simulação da tabela 4.2. $V_{H,min} = -2V$, $V_{H,max} = -4V$, $V_{IPC,min} = -4V$, $V_{IPC,max} = -6V$.	92
III.4	Curvas características de transferência para o TUD OTFT encontradas experimentalmente e modelada pelo MATLAB para primeira simulação da tabela 4.2. $V_{H,min} = -4V$, $V_{H,max} = -6V$, $V_{IPC,min} = -4V$, $V_{IPC,max} = -6V$.	92
III.5	Curvas características de transferência para o TUD OTFT encontradas experimentalmente e modelada pelo MATLAB para segunda simulação da tabela 4.2. $V_{H,min} = -3V$, $V_{H,max} = -5V$, $V_{IPC,min} = -4V$, $V_{IPC,max} = -6V$.	93
III.6	Curvas características de transferência para o TUD OTFT encontradas experimentalmente e modelada pelo MATLAB para terceira simulação da tabela 4.2. $V_{H,min} = -2V$, $V_{H,max} = -4V$, $V_{IPC,min} = -4V$, $V_{IPC,max} = -6V$.	93
III.7	Curvas das funções de ajuste $H(V_{gs})$ e $HH1(V_{gs})$ para o TUD OTFT usando os parâmetros da segunda simulação na tabela 4.2. $V_{H,min} = -3V$, $V_{H,max} = -5V$, $V_{IPC,min} = -4V$, $V_{IPC,max} = -6V$.	94
III.8	Curvas das funções de ajuste $H(V_{gs})$ e $HH1(V_{gs})$ para o TUD OTFT usando os parâmetros da terceira simulação na tabela 4.2. $V_{H,min} = -2V$, $V_{H,max} = -4V$, $V_{IPC,min} = -4V$, $V_{IPC,max} = -6V$.	94
III.9	Curvas das funções de ajuste $IPC(V_{gs})$ e $IIS1(V_{gs})$ para o TUD OTFT usando os parâmetros da segunda simulação na tabela 4.2. $V_{H,min} = -3V$, $V_{H,max} = -5V$, $V_{IPC,min} = -4V$, $V_{IPC,max} = -6V$.	95

III.10	Curvas das funções de ajuste $IPC(V_{gs})$ e $IIS1(V_{gs})$ para o TUD OTFT usando os parâmetros da terceira simulação na tabela 4.2. $V_{H,min} = -2V$, $V_{H,max} = -4V$, $V_{IPC,min} = -4V$, $V_{IPC,max} = -6V$	95
III.11	Curvas das funções de ajuste $IPC(V_{gs})$ e $IIS1(V_{gs})$ para o TUD VOTFT usando os parâmetros da segunda simulação na tabela 4.4. $V_{H,min} = -4V$, $V_{H,max} = -6V$, $V_{IPC,min} = -1.8V$, $V_{IPC,max} = -2.2V$	96
III.12	Curvas das funções de ajuste $IPC(V_{gs})$ e $IIS1(V_{gs})$ para o TUD VOTFT usando os parâmetros da terceira simulação na tabela 4.4. $V_{H,min} = -3.5V$, $V_{H,max} = -5.5V$, $V_{IPC,min} = -1.8V$, $V_{IPC,max} = -2.2V$	96
IV.1	Curvas características de saída para o TUD VOTFT encontradas experimentalmente e modelada pelo MATLAB com $V_{gs} = 0V$, $-2V$, $-4V$ e $-6V$ através do método da menor média usando os parâmetros da tabela 4.6.	97
IV.2	Curvas características de saída para o TUD VOTFT encontradas experimentalmente e modelada pelo MATLAB com $V_{gs} = 0V$, $-2V$, $-4V$ e $-6V$ através do método do menor valor máximo usando os parâmetros da tabela 4.6.	97

LISTA DE TABELAS

3.1	Parâmetros de ajuste da extração para o TUD VOTFT (parte superior da tabela) e TUD OTFT (parte inferior da tabela) para um intervalo com tamanho igual a 2V. Os dados da tabela correspondem às três simulações.	28
3.2	Conjunto de todos os parâmetros mencionados neste projeto. Resumidamente: Parâmetros de ajuste da extração, parâmetros ajustáveis do modelo e parâmetros extraídos do modelo.	29
3.3	Parâmetros genéricos para OTFT's, geralmente encontrados em outros modelos. Na tabela, R corresponde a resistências dos contatos ôhmicos e μ_{fet0} a mobilidade para campos elétricos baixos. Os parâmetros extraídos abaixo para o TUD OTFT e TUD VOTFT são resultados da otimização.	31
4.1	Parâmetros do modelo UMEM extraídos através do MATHCAD e MATLAB para o CINVESTAV OTFT. Esses dados são utilizados para modelar as curvas 4.1, 4.2 e 4.3.	34
4.2	Parâmetros extraídos do TUD OTFT através do MATLAB para diferentes parâmetros de ajuste com intervalos de largura de 2V. Os dados da tabela correspondem às três simulações. A segunda simulação foi considerada como a melhor.	37
4.3	Parâmetros extraídos do TUD OTFT através do MATLAB para diferentes parâmetros de ajuste com intervalos de largura de 1V. Os dados da tabela correspondem às quatro simulações.	38
4.4	Parâmetros extraídos do TUD VOTFT através do MATLAB para diferentes parâmetros de ajuste. Os dados da tabela correspondem às três simulações. A terceira simulação foi considerada como a melhor, para a qual os parâmetros extraídos foram usados em simulações de circuitos no QUCS (seção 4.6).	41
4.5	Parâmetros ajustáveis do modelo para o TUD OTFT encontrados manualmente e otimizados pelo MATLAB através dos métodos da menor média e menor valor máximo. Os parâmetros encontrados manualmente são utilizados para validação do modelo no QUCS (seção 4.5). Os parâmetros encontrados pelo método da menor média foram considerados melhores.	48
4.6	Parâmetros ajustáveis do modelo para o TUD VOTFT encontrados manualmente e otimizados pelo MATLAB através dos métodos de menor média e menor valor máximo. Os parâmetros encontrados manualmente são utilizados para validação do modelo no QUCS (seção 4.5). Os parâmetros encontrados pelo método do menor valor máximo são utilizados nas simulações de circuitos (seção 4.6).	51

LISTA DE SÍMBOLOS E ABREVIACÕES

0.1 Abreviações

Qucs	<i>Quite Universal Circuit Simulator</i>
UMEM	<i>Unified Model and Parameter Extraction Method</i>
MOSFET	<i>Metal Oxide Semiconductor Field Effect</i>
TFT	<i>Thin Film Transistor</i>
OTFT	<i>Organic Thin Film Transistor</i>
VOTFT	<i>Vertical Organic Thin Film Transistor</i>
TUD OTFT	OTFT desenvolvido na Universidade técnica de Dresden
TUD VOTFT	OTFT desenvolvido na Universidade técnica de Dresden
CINVESTAV OTFT	OTFT desenvolvido no instituto CINVESTAV

0.2 Parâmetros físicos dos transistores

W	Largura do canal	[cm]
L	Comprimento do canal	[cm]
d_i	Espessura do dielétrico	[cm]
μ_0	Mobilidade de banda do material	$\text{cm}^2 \text{V}^{-1} \text{s}^{-1}$

0.3 Símbolos dos parâmetros/funções associados à extração do UMEM

m	Parâmetro de ajuste da região do joelho da curva do modelo UMEM	
λ	Modulação do comprimento do canal	[1/V]
α_s	Parâmetro de modulação da tensão de saturação da curva característica de saída do modelo UMEM	
γ_a	Parâmetro de ajuste da mobilidade do canal no modelo UMEM	
V_T	Tensão de limiar	[V]
V_{aa}	Segundo parâmetro de ajuste da mobilidade do canal no modelo UMEM	[V]
R_D	Resistência de contato do dreno	[Ω]
R_S	Resistência de contato da fonte	[Ω]
R	Resistência série dos contatos ôhmicos	[Ω]
μ_{fet}	Mobilidade do canal	$\text{cm}^2 \text{V}^{-1} \text{s}^{-1}$
μ_{fet0}	Mobilidade do canal para baixos valores de V_{gs}	$\text{cm}^2 \text{V}^{-1} \text{s}^{-1}$
g_{chi}	Condutividade do canal na região de triodo profundo	S
$V_{ds,SAT}$	Tensão de saturação para as curvas de saída do UMEM	[V]
$I_{ds,lin}$	Corrente na região linear da curva característica de saída	[A]
$I_{ds,sat}$	Corrente na região de saturação da curva característica de saída	[A]
$H(V_{gs})$	Função de ajuste da extração dos parâmetros V_T , V_{aa} e γ_a	
$HH1(V_{gs})$	Função de ajuste da extração dos parâmetros V_T , V_{aa} e γ_a	
$IIS1(V_{gs})$	Função de ajuste da extração do parâmetro α_s	
$IPC(V_{gs})$	Função de ajuste da extração do parâmetro α_s	
$V_{H,min}$	Parâmetro de ajuste da extração que representa o limite inferior do intervalo usado para $H(V_{gs})$	[V]
$V_{H,max}$	Parâmetro de ajuste da extração que representa o limite superior do intervalo usado para $H(V_{gs})$	[V]
$V_{IPC,min}$	Parâmetro de ajuste da extração que representa o limite inferior do intervalo usado para $IIS1(V_{gs})$	[V]
$V_{IPC,max}$	Parâmetro de ajuste da extração que representa o limite superior do intervalo usado para $IIS1(V_{gs})$	[V]

0.4 Símbolos dos parâmetros/funções associados ao modelo UMEM

V_{ds}	Tensão entre dreno e fonte	[V]
V_{gs}	Tensão entre porta e fonte	[V]
I_{ds}	Corrente total entre dreno e fonte	[A]
I_S	Corrente entre dreno e fonte na região sublimiar	[A]
I_{0S}	Corrente de fuga para curva de transferência de saturação	[A]
I_0	Corrente de fuga para curva de transferência linear	[A]
$\frac{V_{ds}}{CC}$	Fator de correção somado às curvas de saída devido à histerese	
Q	Parâmetro ajustável do modelo que modula a transição entre as regiões sublimiar e de saturação	
DV	Parâmetro ajustável do modelo que modula a aproximação das curvas do UMEM às experimentais na região sublimiar	
S	Parâmetro ajustável do modelo que modula a aproximação das curvas do UMEM às experimentais na região sublimiar	

Capítulo 1

Introdução

1.1 Contextualização

Os TFT's (*Thin film transistors*) foram primeiramente utilizados com o propósito de medir a mobilidade do silício amorfo hidrogenado (a-Si:H). Diversos polímeros e materiais orgânicos foram utilizados como material semicondutor dessas estruturas, que devido à baixa mobilidade, possuíam aplicações limitadas [2]. No entanto, após muitos anos de pesquisa, a mobilidade desses transistores foi aprimorada, superando em muitos casos àquela do TFT de silício amorfo. Com isso, o número de aplicações dessa tecnologia cresceu substancialmente para um custo de produção baixo (um dos processos de fabricação envolve simples impressões desse dispositivos). Animações em telas finas, células fotovoltaicas aplicadas em grandes áreas e flexíveis, implementações em sistemas RFID são algumas das motivações que mantêm grupos de pesquisas acadêmicos e industriais concentrando suas atenções na modelagem e entendimento dos princípios físicos envolvidos nos transistores OTFT (*Organic thin film transistor*). Um dos modelos desenvolvidos, chamado UMEM (*Unified model and parameter extraction method*), foi aplicado em 2001 na extração de parâmetros e modelagem de um a-Si:H TFT. O modelo foi aprimorado em anos posteriores e aplicado à semicondutores orgânicos. Uma introdução ao seu funcionamento será apresentada nos próximos capítulos.

1.2 Definição do problema

Esse projeto é a sucessão do uso de um modelo compacto, UMEM, para extração de parâmetros e simulação de transistores orgânicos de filme fino, OTFT. O projeto anterior consistiu em avaliar a implementação do modelo em um simulador de circuito, QUCS (*Quite universal circuit simulator*), através da extração dos parâmetros, realizada em outra plataforma, MATHCAD. No entanto, devido à uma discrepância encontrada nas simulações realizadas em ambas plataformas, não foi possível utilizar os parâmetros obtidos do UMEM na simulação do transistor (os parâmetros extraídos, então, foram modificados, a fim de obter uma boa aproximação com os dados experimentais). O problema consistiu na determinação da corrente em uma curva de saída, dado

um valor de tensão aplicado na porta do transistor. No simulador de circuitos, QUCS, a corrente encontrada foi 100 vezes menor do que a modelada pelo programa MATHCAD, e a determinada experimentalmente. As possíveis causas dessa discrepância foram atribuídas a diferenças entre os dois códigos utilizados nas duas plataformas.

Com a solução desse problema, é possível verificar a aplicação do UMEM à diferentes tecnologias, como o transistor orgânico vertical de filmes finos, VOTFT (*Vertical organic thin film transistor*), assim como simular o uso desse transistor em diferentes topologias tradicionais de circuitos.

1.3 Objetivos do projeto

O objetivo desse projeto é identificar as causas da discrepância obtida na implementação do modelo entre as duas plataformas, assim como aplicá-lo à uma tecnologia em desenvolvimento, conhecida como transistores orgânicos verticais de filmes finos, VOTFT. Finalmente, caso a aproximação com os dados experimentais seja satisfatória, topologias de circuito serão simuladas com o objetivo de verificar o desempenho do transistor para tensões de dreno e porta diferentes daquelas aplicadas experimentalmente.

1.4 Apresentação do relatório

No capítulo 2 é feita uma revisão bibliográfica sobre transistores orgânicos verticais de filme fino, modelos compactos, modelagem e extração de parâmetros utilizando o UMEM. Ainda nesse capítulo, as topologias de circuito utilizadas no projeto são introduzidas, assim como a linguagem de programação e o simulador utilizados. Em seguida, o capítulo 3 descreve todos procedimentos desenvolvidos, incluindo a substituição do código implementado no MATHCAD por um equivalente em MATLAB, o ajuste do código em Verilog A, a aplicação do UMEM ao VOTFT, além de simulações do último em diferentes circuitos. Os resultados das simulações, assim como parâmetros extraídos são apresentados no capítulo 4, seguido das conclusões no capítulo 5. O anexo contém o código em Verilog A atualizado, o algoritmo de extração e modelo implementado no MATLAB, além de alguns resultados das simulações.

Capítulo 2

Revisão Bibliográfica

2.1 Introdução

Este capítulo introduz os tópicos necessários para entendimento do projeto. Inicialmente, conceitos relativos aos VOTFT's, modelos compactos, incluindo o UMEM em uma seção específica, são apresentados. Em seguida, sucintamente, explica-se as características principais dos amplificadores inversores, assim como um dos softwares (QUCS + Verilog A) utilizados.

2.2 Transistores verticais de filmes finos orgânicos (VOTFT)

Transistores orgânicos de filme fino (OTFT) têm sido cada vez mais utilizados, principalmente em aplicações que demandem estruturas flexíveis e área considerável, como observado nas tecnologias envolvendo telas flexíveis e peles artificiais. No entanto, tais dispositivos possuem performance bastante limitada quando comparados aos MOSFET's. Devido a baixa mobilidade dos OTFT's (maior registrada foi igual à $25 \text{ cm}^2 \text{ V}^{-1} \text{ s}^{-1}$), esses transistores não podem ser usados em aplicações de alta frequência de comutação. Outra desvantagem é fato do comprimento do canal não poder ser encurtado para uma escala nanométrica, dimensão usada atualmente em circuitos integrados de microprocessadores, devido a limitações impostas pelo aumento das resistência dos contatos. No entanto, é uma tecnologia de baixo custo, que também pode ser usado para implementar osciladores, usados em sistemas RFID (Identificação por radio-frequência) [3].

O transistor vertical de filme fino orgânico (VOTFT) pode ser usado para minimizar o problema dos transistores orgânicos em relação ao comprimento do canal. Devido a sua estrutura vertical, usada para o fluxo de cargas, o comprimento efetivo do canal pode ser reduzido à espessura da camada orgânica, como pode ser visto na figura 2.1. Diferentes tecnologias de VOTFT's são discutidas nas subseções seguintes.

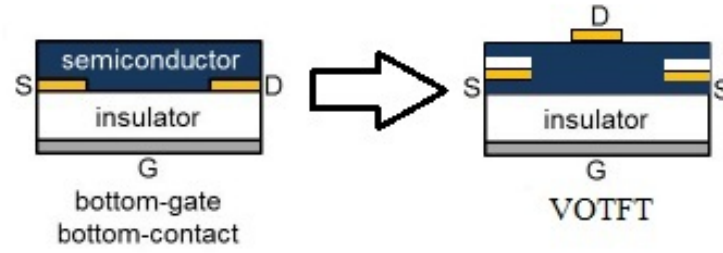


Figura 2.1: Arquitetura de um transistor vertical vista a partir de uma das arquiteturas usadas para OTFT's. O fluxo de cargas ocorre a partir dos dois eletrodos correspondentes à fonte em direção ao dreno, verticalmente.

2.2.1 Transistores orgânicos de base permeável (OPBT)

Essa tecnologia se assemelha ao transistor de junção bipolar (TBJ), no entanto com uma estrutura de condução vertical. Pode ser analisado como uma associação entre três terminais através de dois diodos com sentidos de polarização opostos, como pode ser visto na figura 2.2. Ele é constituído por duas camadas de material orgânico, separados por um eletrodo metálico correspondente a base. Os eletrodos correspondentes ao emissor e coletor, também metálicos, se localizam no topo e abaixo do transistor, respectivamente.

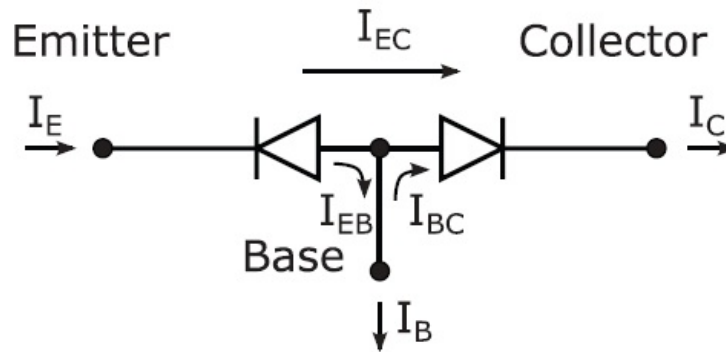


Figura 2.2: Diagrama correspondente ao princípio de funcionamento de um OPBT.

Quando o diodo entre a base e o emissor é diretamente polarizado, uma corrente é injetada no terminal do emissor. Apesar do sentido da corrente se opor ao sentido de condução do diodo, essa corrente atravessa a base, que é muito fina, em direção ao coletor, criando uma componente I_{EC} . Como nem toda corrente é transmitida, uma parte dela, I_{EB} é drenada pelo terminal da base. A corrente total transmitida ao coletor pode ser considerada igual, então, a soma entre as correntes injetada na base em direção ao coletor I_{BC} e a corrente transmitida do emissor I_{EC} . Como a corrente da base I_B é controlada pela tensão de polarização do diodo V_{BE} , diz-se que a corrente transmitida ao coletor I_C é modulada pela corrente injetada na base. Assim, define-se o coeficiente de transmissão α como sendo igual a razão entre a corrente transmitida do emissor ao coletor em relação a corrente total injetada no emissor, como pode ser visto na equação (2.1). Algumas formas de aumentar o coeficiente α foram desenvolvidas, tais como o transporte de cargas "quentes" através de uma fina camada metálica (base) e a transmissão através de pequenos espaços

no eletrodo da base. A figura 2.3 mostra arquitetura do dispositivo.

$$\alpha = \frac{I_{EC}}{I_E} . \quad (2.1)$$

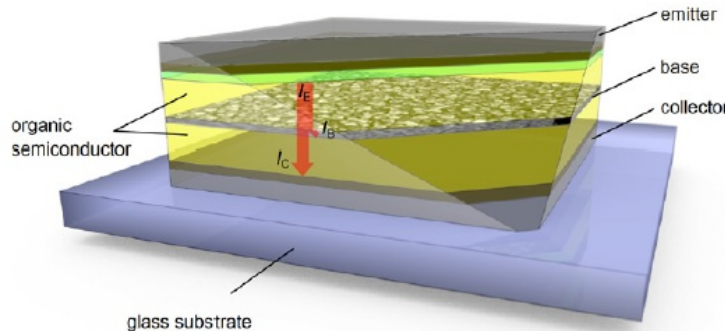


Figura 2.3: Arquitetura correspondente ao OPBT.

2.2.2 Transistores orgânicos de indução estática (OSIT)

Essa tecnologia possui a mesma arquitetura dos OPBT's, mencionados na seção anterior. A diferença entre os dois dispositivos está na forma como o eletrodo da base é manufaturado. No OSIT, ele é obtido a partir do uso de uma máscara de esferas de polistireno, que são depositadas sobre o eletrodo do emissor (neste caso localizado na parte baixa do transistor). Acima da máscara, o eletrodo de base é colocado e, em seguida, as esferas são removidas com o uso de uma fita adesiva, resultando em uma base com maior permeabilidade. Sobre o eletrodo de base, cujos orifícios possuem diâmetro entre 100 nm e 200 nm, é depositado a camada do componente orgânico e acima deste o eletrodo do coletor. Há ainda finas camadas de isolante separando os eletrodos entre si e do material orgânico. O resultado deste procedimento é um aumento do coeficiente de transmissão α e consequentemente a amplificação da corrente. Esse dispositivo também apresenta razão ON/OFF maior e o uso de isolantes nas proximidades da base reduz as correntes de fuga. A figura 2.4 mostra o processo de fabricação deste dispositivo.

2.2.3 Transistores orgânicos de barreira Schottky (OSBT)

A principal diferença desta tecnologia das demais é que o terminal de controle (porta) é separado do canal de condução. A estrutura restante, chamada de diodo Schottky comutável, é então composta pela célula ativa, contendo dreno e fonte, cuja condução é principalmente controlada pela resistência de contato nesses terminais, ao invés da condutividade do canal. A figura 2.5 mostra a arquitetura desse transistor. Observa-se que como o terminal de porta está separado do semicondutor pelo eletrodo da fonte, é necessário que este último seja permeável a campos elétricos DC, de forma que a porta possa afetar a condução no canal.

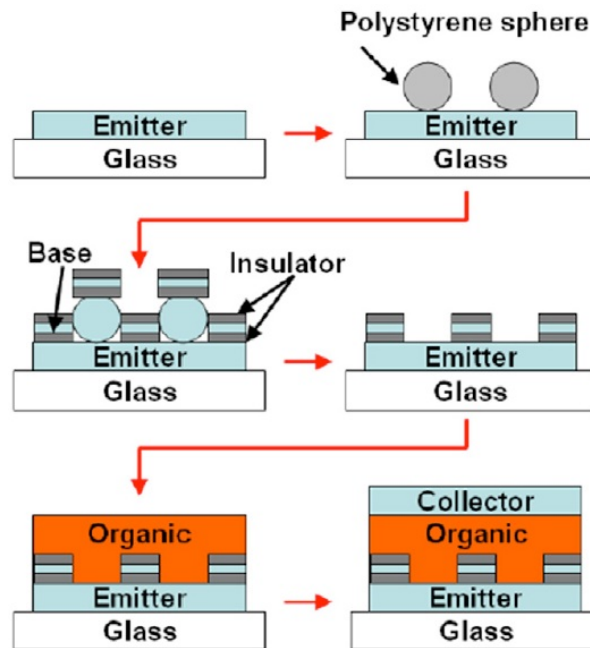


Figura 2.4: Procedimentos para construção de um OSIT.

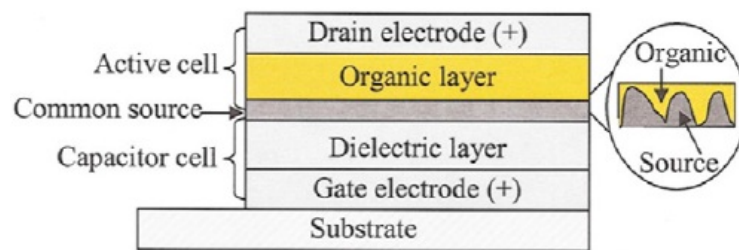


Figura 2.5: Arquitetura correspondente ao OSBT.

2.2.4 Transistores orgânicos de efeito de campo (VOFET)

Esses transistores combinam transporte horizontal e vertical de carga. Antes dos portadores de carga se deslocarem verticalmente (comprimento efetivo do canal), eles fluem na interface do dielétrico da porta nas proximidades da fonte. Tal fator permite que as propriedades dos transistores de efeito de campo sejam conservadas, ao mesmo passo que o canal é encurtado, devido a condução de cargas no sentido vertical. Esse foi a arquitetura do transistor usado neste projeto, a qual pode ser vista na figura 2.6.

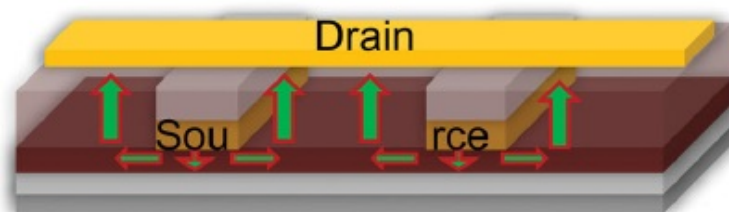


Figura 2.6: Arquitetura correspondente ao VOFET usado neste projeto.

2.3 Modelos compactos para OTFT's

De acordo com [4], um modelo compacto é uma forma de descrever matematicamente o funcionamento de um dispositivo de forma prática, capturando somente os aspectos físicos mais importantes. Com o objetivo de reproduzir as características elétricas de OTFT's, modelos compactos foram desenvolvidos, através da combinação de parâmetros físicos e específicos de cada modelo. No entanto, devido a grande variedade de arquiteturas e materiais emergentes, não é possível usar nenhum dos modelos seguramente, pois ainda há muitos efeitos físicos desconhecidos dessas estruturas. Assim, o modelo para MOSFET's tem sido bastante usado em OTFT's. No entanto, os dois transistores possuem diferentes comportamentos físicos e por isso não podem ser analisados pelo mesmo modelo. Em OTFT's, as seguintes considerações podem ser feitas:

1 - Estrutura molecular desordenada dos materiais policristalinos orgânicos (domínios com tamanhos e formatos diferentes), o que resulta em aumento do número de armadilhas e maior dificuldade em modelar o transporte de carga. Efeitos de localização de carga implica em uma dependência diferente da mobilidade em relação à tensão V_{gs} .

2 - Junções metal/semicondutor formadas nas conexões dos eletrodos correspondentes ao dreno e a fonte com o filme orgânico. Como nos MOSFET's, as regiões próximas a esses contatos é altamente dopada, não há problemas com injeção de carga. No entanto, em OTFT's a resistência dos contatos (chamada resistência serial) é alta e muitas vezes maior do que a resistência do canal.

Além disso, OTFT's não possuem relações perfeitamente lineares para as curvas características $I_{ds,lin}xV_{gs}$ e $\sqrt{I_{ds,sat}}xV_{gs}$.

Com o objetivo de modelar as principais características físicas dos transistores de filmes finos orgânicos, vários modelos foram desenvolvidos, geralmente com o mesmo nível de descrição. Entre eles, pode-se citar Marinov et al, em [5], Li et al, em [6] e Estrada et al, em [7], [8] e [9]. As principais características de cada um deles é mostrada nos parágrafos que seguem:

Marinov et al: Modelo que leva em consideração o comportamento exponencial da corrente na região sublimiar, assim como a dependência da mobilidade em relação a V_{gs} por um função de potência através de um parâmetro específico γ . O cálculo da mobilidade é feito espacialmente, logo variações da tensão ao longo do canal são incluídas. Esse modelo, diferentemente dos demais, também considera o efeito de distorção das curvas de transferências para altos valores de V_{gs} , frequentemente encontrado em OTFT's.

Li et al: Modelo dependente de parâmetros empíricos, cujos valores são estimados através de várias simulações numéricas e modificações simultâneas. Esse modelo, no entanto, não faz distinções entre as regiões sublimiar e acima da tensão limiar (todos dois estados representados por um mesma equação). O resultado disso é que a estimação da tensão de saturação não é apropriada e precisa ser corrigida através de parâmetros de ajuste do modelo.

Estrada et al: Modelo usado neste projeto, que será descrito com detalhes na próxima seção.

Todos os modelos citados anteriormente apresentam ajuste satisfatório às curvas experimentais, quando avaliados para transistores que se comportam idealmente. No entanto, ainda há muitos

efeitos físicos a serem levados em consideração em um modelo compacto, os quais serão citados nos próximos parágrafos.

O fato dos OTFT's operarem basicamente no regime de acumulação leva a alterações dos valores estimados para tensão limiar em relação aos MOSFET's. Outros fatores também influenciam na determinação de V_T , tais como: As armadilhas na estrutura policristalina, que aumenta a tensão na porta necessária para formação de uma camada de acumulação de portadores de carga e a posição dos eletrodos de fonte e dreno, que alteram a distância necessária que as cargas devem atravessar antes de atingir a interface do semicondutor.

Há também uma modelagem inapropriada da região sublimiar, que somente considera a dependência exponencial da corrente em relação a tensão V_{gs} . OTFT's apresentam também uma parcela correspondente a corrente de deriva dos portadores de carga do corpo.

Em relação a mobilidade, não é considerada sua dependência em relação a tensão V_{ds} na maioria dos modelos, causada pela estrutura desordenada dos semicondutores orgânicos.

A figura 2.7 contém as diferentes arquiteturas de OTFT's, as quais são também determinantes nas características físicas dos dispositivos. No entanto, a maioria dos modelos ainda não considera essas diferenças, possivelmente porque os tornaria muito descritivos, dificultando a implementação computacional.

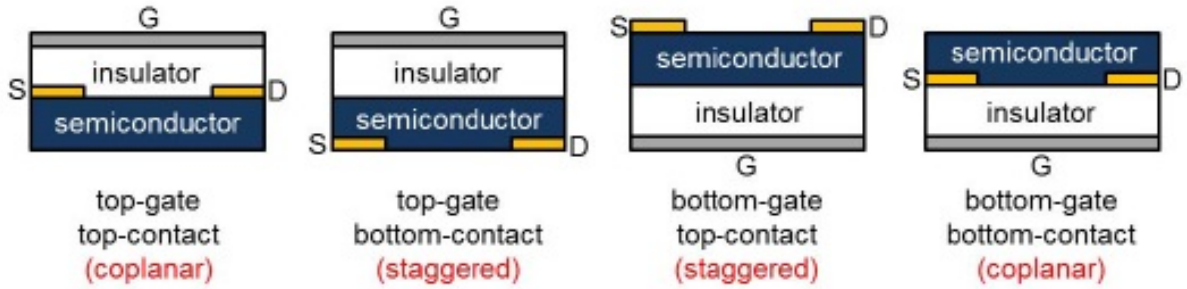


Figura 2.7: Diferentes arquiteturas para um OTFT. Suas denominações são feitas de acordo com a posição relativa dos contatos em relação a camada do semicondutor. Nas estruturas coplanares, dreno e fonte estão no mesmo plano com o canal, enquanto que nas estruturas escalonadas, o canal está separado dos eletrodos por extensões do semicondutor. Para estruturas escalonadas, o aumento da área de contato dos eletrodos reduz a resistência na injeção de portadores.

2.4 O modelo e método de extração unificados (UMEM)

O UMEM, proposto em [8], [7] e [9], é um modelo compacto para OTFT's, que também oferece a possibilidade de extração de parâmetros. O algoritmo de extração possui como entradas duas curvas de transferência $I_{ds} \times V_{gs}$ e uma curva de saída $I_{ds} \times V_{ds}$. Os parâmetros obtidos podem ser usados como entradas das equações do modelo em uma mesma plataforma, produzindo outras curvas de saída e transferência para tensões V_{ds} e V_{gs} diferentes das obtidas experimentalmente. Assim, é possível atingir aproximadamente o mesmo ajuste às curvas experimentais,

independentemente da tensão de polarização.

O modelo com extração unificada não é exclusivo do UMEM. A vantagem do UMEM, no entanto, é não possuir muitos parâmetros que, durante a extração, requerem otimizações computacionais e análises gráficas, aumentando não somente o tempo gasto com simulações, mas também os erros causados por falta de acurácia. Esses parâmetros, muitas vezes, não possuem significado físico e somente são utilizados para estimação de outros parâmetros do modelo utilizado.

É possível, então, aplicar o UMEM à diversas tecnologias de transistores orgânicos, o qual considera, durante a extração e modelagem, características físicas específicas desses TFT's, tais como o efeito do maior número de armadilhas em materiais policristalinos, o que resulta em uma dependência diferenciada da mobilidade em relação a tensão V_{gs} . Há efeitos também relacionados a histerese em OTFT's, tais como as correntes de fuga, que dependem da polarização do dispositivo. E, para alguns casos, distorções na origem das curvas de saída causadas por contatos não-ôhmicos no dreno e na fonte dos transistores.

2.4.1 Algoritmo de extração do UMEM

Todos os parâmetros do UMEM são extraídos para a região acima da tensão limiar, a partir da equação da corrente definida para esse modelo, a qual pode ser vista em (2.2). Essa equação considera a dependência da mobilidade de transistores orgânicos com campos elétricos transversais (polarização da porta V_{gs}), como indicado pela equação (2.3), assim como o efeito da alta resistência dos contatos ôhmicos na condutividade intrínseca do canal de OTFT's. A associação entre as equações (2.3) e (2.4), correspondente a condutividade do canal para um FET na região de triodo profundo (tensões V_{ds} baixas), resulta na condutividade para OTFT's nessa região, como pode ser visto na equação (2.5). O produto entre (2.5) e V_{ds} resulta na expressão da corrente para a região linear, em (2.6), a qual não considera os efeitos dos contatos ôhmicos, o comprimento de modulação do canal ($\lambda = 0$, $R = 0$) e não depende de m ($V_{ds} \ll V_{ds,SAT}$). A equação (2.2) pode ser entendida como a generalização para quaisquer valores de tensão V_{ds} , esteja o transistor operando na região linear ou não. Os parâmetros são, então, extraídos a partir das simplificações para a equação da corrente, na região linear, em (2.6), e na região de saturação, em (2.7). Assim como na equação (2.6), a expressão da corrente para região de saturação também não considera os efeitos dos contatos ôhmicos, a modulação do comprimento do canal ($\lambda = 0$, $R = 0$) e não depende do parâmetro m ($V_{ds} \gg V_{ds,SAT}$).

$$I_{ds} = \frac{\frac{K}{(V_{aa})^{\gamma_a}}}{1 + R \cdot \frac{K}{(V_{aa})^{\gamma_a}} \cdot (V_{gs} - V_T)^{1+\gamma_a}} \times \frac{(V_{gs} - V_T)^{1+\gamma_a} \cdot V_{ds} (1 + \lambda \cdot V_{ds})}{\left[1 + \left[\frac{V_{ds}}{V_{ds,SAT}}\right]^m\right]^{\frac{1}{m}}}, \quad (2.2)$$

$$\mu_{fet} = \mu_0 \left[\frac{(V_{gs} - V_T)}{V_{aa}} \right]^{\gamma_a}, \quad (2.3)$$

$$g_{chi} = \frac{W}{L} C_i \mu_{fet} (V_{gs} - V_T), \quad (2.4)$$

$$g_{\text{chi}} = \frac{W}{L} C_i \frac{\mu_0}{(V_{\text{aa}})^{\gamma_a}} (V_{\text{gs}} - V_{\text{T}})^{1+\gamma_a} , \quad (2.5)$$

$$I_{\text{ds,lin}} = \frac{K}{(V_{\text{aa}})^{\gamma_a}} (V_{\text{gs}} - V_{\text{T}})^{1+\gamma_a} V_{\text{ds}} , \quad (2.6)$$

$$I_{\text{ds,sat}} = \frac{K}{(V_{\text{aa}})^{\gamma_a}} \alpha_s (V_{\text{gs}} - V_{\text{T}})^{2+\gamma_a} , \quad (2.7)$$

Na equação (2.2) pode-se identificar todos os parâmetros do modelo. O fator de acoplamento K , mostrado na equação (2.8), é igual ao produto entre a razão de aspecto do transistor $\frac{W}{L}$, a capacitância da porta C_i e a mobilidade de banda μ_0 , que depende do material, mas é considerada igual à $1 \text{ cm}^2 \text{ V}^{-1} \text{ s}^{-1}$ na maioria dos casos. Os parâmetros V_{aa} e γ_a são parâmetros de ajuste das curvas que modelam a variação da mobilidade com a tensão na porta, como pode ser visto na equação (2.3). A associação entre V_{aa} , γ_a e μ_0 através da equação (2.9) é definida como a mobilidade para campos elétricos transversais baixos. Os parâmetros m e λ não possuem sentido físico e são também usados para ajustar as curvas aos dados experimentais. Enquanto m modela a suavidade da curva no joelho, entre as regiões linear e de saturação, λ está associado à modulação do comprimento do canal e descreve a variação da condutividade do canal com V_{ds} na região de saturação. O ultimo parâmetro, α_s , é utilizado para modelar a tensão de saturação $V_{\text{ds,SAT}}$ através da equação (2.10).

$$K = \frac{W}{L} \mu_0 C_i , \quad (2.8)$$

$$\mu_{\text{fet0}} = \frac{\mu_0}{(V_{\text{aa}})^{\gamma_a}} = \frac{1}{(V_{\text{aa}})^{\gamma_a}} \text{ cm}^2 \text{ V}^{-1} \text{ s}^{-1} , \quad (2.9)$$

$$V_{\text{ds,SAT}} = \alpha_s (V_{\text{gs}} - V_{\text{T}}) , \quad (2.10)$$

Encontra-se os parâmetros V_{T} e γ_a através do método integral, definido em [8], com a corrente de saída definida para operação do transistor na região linear. Ou seja, para esse estágio, aplica-se a equação (2.11) em I_{ds} obtido da curva de transferência experimental para uma baixa tensão de polarização V_{ds} . O parâmetro V_{T} é determinado, então, a partir do ponto de interseção entre o eixo das tensões V_{gs} e a função $H(V_{\text{gs}})$. Similarmente, o parâmetro γ_a é encontrado através do coeficiente angular da reta $H(V_{\text{gs}})$.

$$H(V_{\text{gs}}) = \frac{\int_0^{V_{\text{gs}}} I_{\text{ds}}(x) dx}{I_{\text{ds}}(V_{\text{gs}})} = \frac{1}{2 + \gamma_a} (V_{\text{gs}} - V_{\text{T}}) . \quad (2.11)$$

A curva de transferência obtida a partir de uma baixa tensão de polarização aplicada no dreno ($V_{\text{ds}} \ll V_{\text{gs,MAX}}$) garante a operação do transistor na região linear em um trecho significativo da tensão de varredura V_{gs} . O parâmetro V_{aa} também é encontrado a partir dessa curva. O coeficiente angular S_1 da equação (2.13), obtida do rearranjo dos termos de (2.6) (expressão

para corrente na região linear), pode ser utilizada na equação (2.14) para determinar V_{aa} . Assim, obtém-se os três parâmetros necessários para modelar a mobilidade de um OTFT para qualquer tensão aplicada na porta V_{gs} . O ultimo parâmetro determinado a partir da curva de transferência na região linear é a resistência serial dos contatos ôhmicos $R = R_D + R_S$. Substitui-se na equação (2.12) o valor de I_{ds} medido para a máxima tensão V_{gs} da curva experimental, além dos demais parâmetros já determinados. Em seguida, rearranjando os termos da equação, é possível obter R . Nota-se que a equação (2.12) corresponde à expressão da corrente na região linear, em (2.6), considerando-se a resistência serial, que foi desprezada anteriormente com o propósito de evitar o uso de cálculo numérico na estimação dos parâmetros.

$$I_{ds} = \frac{\frac{K}{(V_{aa})^{\gamma_a}} \cdot (V_{gs} - V_T)^{1+\gamma_a} \cdot V_{ds}}{1 + R \cdot \frac{K}{(V_{aa})^{\gamma_a}} \cdot (V_{gs} - V_T)^{1+\gamma_a}} , \quad (2.12)$$

$$[I_{ds,lin}(V_{gs})]^{\frac{1}{1+\gamma_a}} = \left[\frac{K}{(V_{aa})^{\gamma_a}} V_{ds} \right]^{\frac{1}{1+\gamma_a}} \cdot (V_{gs} - V_T) = S_l (V_{gs} - V_T) , \quad (2.13)$$

$$V_{aa} = \left[\frac{K V_{ds}}{(S_l)^{1+\gamma_a}} \right]^{\frac{1}{\gamma_a}} . \quad (2.14)$$

Para a curva de transferência na região de saturação ($V_{ds} \geq V_{gs,MAX}$), determina-se o parâmetro de modulação da tensão de saturação α_s a partir do cálculo do coeficiente angular S_s da equação (2.15), definida a partir do rearranjo dos termos da equação (2.7). O coeficiente angular é então usado para determinar α_s com o uso da equação (2.16), dado que os demais parâmetros γ_a e V_{aa} já são conhecidos.

$$[I_{ds,sat}(V_{gs})]^{\frac{1}{2+\gamma_a}} = \left[\frac{K}{(V_{aa})^{\gamma_a}} \alpha_s \right]^{\frac{1}{2+\gamma_a}} (V_{gs} - V_T) = S_s (V_{gs} - V_T) , \quad (2.15)$$

$$\alpha_s = \frac{(S_s)^{2+\gamma_a} (V_{aa})^{\gamma_a}}{K} , \quad (2.16)$$

Para determinação de m e λ usa-se uma das curvas de saída experimentais. Os termos da equação (2.2) são rearranjados para se isolar m . Negligenciando-se λ e R , encontra-se a equação (2.17). No entanto, nota-se que a equação (2.17) depende de V_{gs} e $V_{ds,SAT}$, os quais variam de acordo com a curva de saída utilizada. Então, escolhe-se uma tensão de polarização V_{gs} próxima a máxima tensão $V_{gs,MAX}$ medida e que corresponda a curva que se deseja modelar. Para determinação do parâmetro λ , usa-se a mesma curva de saída na equação (2.18). Para essa ultima equação, define-se $V_{ds,2}$ e $I_{ds,2}$ como a maior tensão e corrente medidas para a curva escolhida, respectivamente.

$$m = \log 2 / \log \left[\frac{K \cdot \mu_{FET}(V_{gs}) \cdot \alpha_s \cdot (V_{gs} - V_T)}{[1 + K \cdot \mu_{FET}(V_{gs}) \cdot (V_{gs} - V_T)] \cdot I_{ds,SAT} \cdot (V_{ds,SAT})} \right] , \quad (2.17)$$

$$\lambda = \left\{ \frac{\frac{(I_{ds,2})}{(V_{ds,2})^2} \cdot [1 + R \cdot K \cdot \mu_{FET}(V_{gs}) \cdot (V_{gs} - V_T)] \cdot \left[1 + \left[\frac{(V_{ds,2})}{\alpha_s \cdot (V_{gs} - V_T)}\right]^m\right]^{\frac{1}{m}}}{K \cdot \mu_{FET}(V_{gs}) \cdot (V_{gs} - V_T)} \right\} - \frac{1}{V_{ds,2}}. \quad (2.18)$$

Uma observação importante é que a função $H(V_{gs})$ não é perfeitamente linear. Logo, como vários intervalos V_{gs} podem ser escolhidos para o cálculo da integral na equação (2.11), os parâmetros extraídos V_T e γ_a podem variar de acordo com região do gráfico $H(V_{gs})$ escolhida. Semelhantemente, a equação (2.15) não é linear, pois a curva $I_{ds}xV_{gs}$ utilizada para determiná-la também não é linear (curva de transferência na região de saturação). Ou seja, para encontrar α_s , deve-se escolher um intervalo em que a curva da equação (2.15) seja linear. Por esse motivo, $H(V_{gs})$ e $[I_{ds}(V_{gs})]^{\frac{1}{\gamma_a+2}}$ são chamadas funções de ajuste, pois sua determinação depende do ajuste às curvas experimentais obtido para um dado conjunto dos parâmetros extraídos.

O modelo e a extração de parâmetros para os efeitos dos contatos não-ôhmicos podem ser encontrados em [7]. Como a região próxima a origem das curvas de saída experimentais dos OTFT's usados neste projeto eram lineares, esses efeitos não foram considerados.

2.4.2 Modelo UMEM

Após a determinação dos parâmetros do UMEM com o uso do algoritmo descrito na seção anterior, é possível utilizá-los na equação da corrente de saída definida em (2.2) para modelar tanto a região acima, quanto abaixo da tensão limiar dos OTFT's.

Como os OTFT's geralmente apresentam valores altos para a razão ON/OFF na região sublimiar, a corrente para $V_{gs} < V_T$ é representada com dependência exponencial em relação a tensão aplicada na porta, como pode ser visto na equação (2.19). Nessa equação, define-se I_S como a corrente modelada na região sublimiar, I_{ds} como a corrente da equação (2.2) e $V_T + DV$ como a tensão aplicada na porta para a qual a corrente começa a crescer exponencialmente na região sublimiar. O parâmetro S representa o ajuste entre as curvas modeladas e experimentais para $V_{gs} < V_T$ e pode ser calculado através da inclinação da curva de transferência experimental na região sublimiar em um gráfico mono-log.

$$I_S = I_{ds}(V_T + DV, V_{ds}) \cdot e^{\frac{2.3}{S} \cdot (V_{gs} - V_T)}, \quad (2.19)$$

A corrente total, considerando as componentes acima e abaixo da tensão limiar, pode ser representada pela equação (2.22). É importante notar o efeito da função hiperbólica. Enquanto a componente da região sublimiar decresce com o aumento da tensão V_{gs} (o termo $1 - \tanh$ se torna menor), a componente para $V_{gs} > V_T$ cresce (o termo $1 + \tanh$ se torna maior). As funções hiperbólicas, então, funcionam como termos de modulação da corrente. Os parâmetro Q na equação (2.22) modela a transição entre as regiões abaixo e acima de V_T . O termo I_0 representa a corrente de fuga, que varia de acordo com a tensão de polarização V_{ds} e pode ser determinado a partir das curvas experimentais para $V_{gs} \ll V_T$. Para as curvas de saída, soma-se a I_{ds} um termo

de correção $\frac{V_{ds}}{CC}$, mostrado na equação (2.23), correspondente as diferenças causadas pelos efeitos da histerese entre as curvas de saída e transferência. O valor de CC pode ser encontrado como o inverso da inclinação da curva de saída para $V_{gs} = 0$.

$$I_{ds,0} = I_S \cdot \frac{1}{2} \cdot [1 - \tanh(Q \cdot (V_{gs} - (V_T + DV)))] , \quad (2.20)$$

$$I_{ds,1} = I_{ds} \cdot \frac{1}{2} \cdot [1 + \tanh(Q \cdot (V_{gs} - (V_T + DV)))] , \quad (2.21)$$

$$I_{ds} = I_{ds,0} + I_{ds,1} + I_0 , \quad (2.22)$$

$$I_{ds} = I_{ds,0} + I_{ds,1} + \frac{V_{ds}}{CC} . \quad (2.23)$$

É importante notar também que os parâmetros Q , DV e S possuem equivalentes para a curva de transferência na região de saturação, iguais à QS , DVS e SS . Enquanto Q e QS são geralmente iguais, DVS e SS podem ser diferentes de DV e S , devido a efeitos da histerese nos OTFT's. No entanto, para este projeto, por motivos de simplificação, eles foram considerados iguais.

Nota-se também que os efeitos da histerese são intrínsecos de cada experimento. Por exemplo, a corrente de fuga I_0 é encontrada somente nas curvas de transferência e dependem de como a tensão V_{gs} é excursionada ($ON > OFF$ ou $OFF > ON$). Ou seja, I_0 possivelmente nunca será o mesmo se medido em diversos experimentos, o que justifica a necessidade de ser desprezado em simulações de circuitos, já que as condições de polarização não são as mesmas também. O mesmo pode ser dito em relação aos demais efeitos da histerese, tais quais as diferenças entre DV e DVS , S e SS e a soma do fator de correção $\frac{V_{ds}}{CC}$ às curvas modeladas. No entanto, como CC geralmente tende ao infinito (Correntes para $V_{gs} = 0$ geralmente são nulas), a consideração desse fator nas simulações pouco altera os resultados.

2.5 Amplificadores inversores

Esses amplificadores são geralmente utilizados no estágio de ganho em um circuito integrado [10]. São chamados de inversores pois a saída do sinal AC é defasada em 180° em relação a entrada. A topologia descrita a seguir representa a configuração básica fonte comum, na qual somente um transistor é usado em série com um resistor de carga R_D , conforme mostrado na figura 2.8.

Essa configuração é chamada fonte comum pois a fonte é vista como terminal comum para os acessos de saída, entre dreno e fonte, e de entrada, entre porta e fonte, do transistor [11]. Esse circuito pode ser analisado através da associação entre as equações da corrente para o transistor e para a carga resistiva R_D . Aplicando-se tensões de entrada entre $0V$ e V_{DD} no transistor, obtém-se a curva de transferência estática, na figura 2.9, dessa topologia, que relaciona V_{gs} e V_{ds} a partir da interseção entre as curvas $I_{ds}xV_{ds}$ do transistor e a curva de carga $I_{ds} = \frac{V_{DD}}{R_D} - \frac{V_{ds}}{R_D}$. Nessa

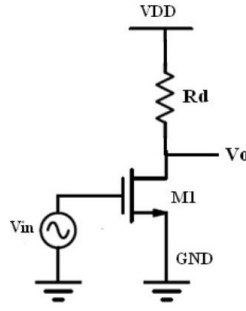


Figura 2.8: Topologia básica correspondente à um amplificador inversor fonte comum.

configuração, o amplificador de transcondutância é substituído por um de tensão e a amplificação ocorre linearmente em torno de um ponto quiescente escolhido entre o corte e a região de triodo do transistor. Nesse ponto, o ganho da topologia pode ser determinado através da equação (2.24), em que g_m e r_o são a transcondutância e a resistência de saída do transistor, respectivamente, para o ponto de operação escolhido. A resistência de saída do amplificador pode ser vista na equação (2.25).

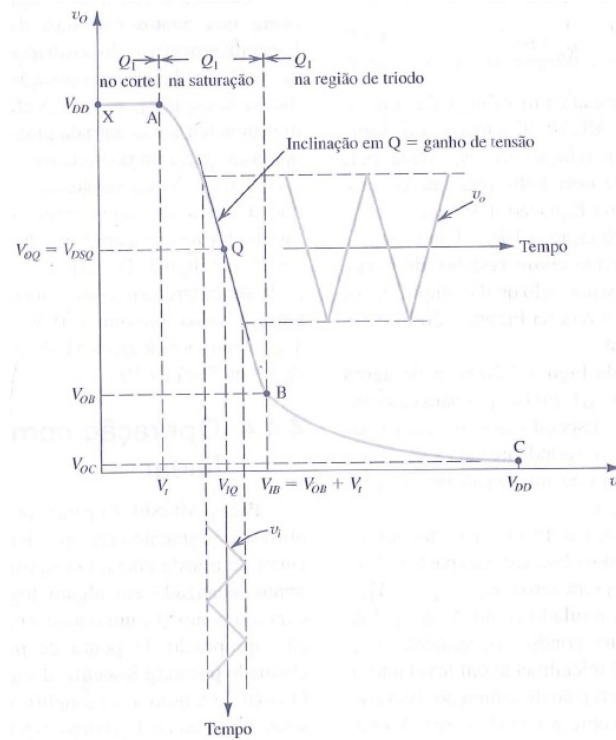


Figura 2.9: Curva de transferência de tensão de um amplificador inversor fonte comum.

$$\frac{V_{out}}{V_{in}} = -g_m \cdot (r_o \parallel R_D) , \quad (2.24)$$

$$R_{out} = (r_o \parallel R_D) , \quad (2.25)$$

2.5.1 Amplificadores inversores CMOS

Esses amplificadores utilizam a mesma topologia do circuito descrito acima. No entanto, a carga resistiva é substituída por um componente ativo. Esse componente pode ser um transistor com porta e dreno conectados entre si (carga ativa) ou um transistor polarizado por outra fonte de tensão (fonte de corrente). Há também a configuração push-pull, que apesar da curva de carga ser obtida de forma diferente, também defasa o sinal de entrada em 180° .

2.5.1.1 Carga ativa

Essa topologia apresenta o menor ganho de tensão dentre as três descritas nesta seção. Na figura 2.10, utiliza-se como carga um transistor tipo P e como fonte comum um transistor tipo N. Observa-se que a conexão na forma de diodo-MOS mantém o transistor tipo P em saturação. A partir da varredura da tensão de entrada V_{in} do circuito, é possível obter a característica estática de transferência de tensão desta topologia na figura 2.10. Com a conexão entre porta e dreno do transistor M2, suas curva de saída e transferência se tornam iguais. Assim, como pode ser observado em 2.10, M2 começa a conduzir quando $V_{gs,2} < V_{Tp}$, o que implica em $V_{ds,2} < V_{Tp}$ ou $V_{out} - V_{DD} < -\alpha$, em que $\alpha = |V_{Tp}|$ do transistor tipo P. Isso limita a excursão do sinal de saída à um valor máximo de $V_{DD} - V_{Tp}$, o qual é atingido quando $V_{in} = V_{Tn}$ no transistor tipo N.

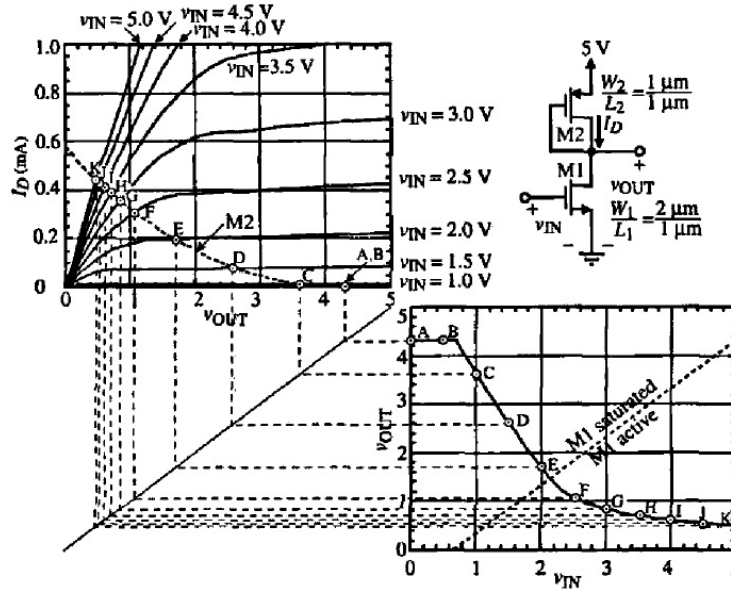


Figura 2.10: Curva de transferência de tensão de um amplificador inversor com carga ativa.

O modelo de pequenos sinais dessa topologia pode ser encontrado na figura 2.11, para o qual determina-se o ganho e a resistência de saída, mostrados nas equações (2.26) e (2.27), respectivamente. Com a inclusão das capacitâncias parasitas, determina-se o ganho no domínio da frequência a partir da equação (2.28), cujo o polo, determinante da banda do sinal de saída (zero possui magnitude maior que a do polo), pode ser encontrado através da equação (2.29). A magnitude do polo determina para qual frequência o ganho do circuito começa a ser reduzido em 20dB por década. Nota-se que a redução da resistência de saída (redução do ganho), através do deslocamento do

ponto de operação, aumenta a frequência de corte, mantendo o produto ganho-banda constante, conforme esperado.

$$\frac{V_{out}}{V_{in}} = -\frac{g_{m1}}{g_{m2}}, \quad (2.26)$$

$$R_{out} = \frac{1}{g_{m2} + g_{ds1} + g_{ds2}} \approx \frac{1}{g_{m2}}, \quad (2.27)$$

$$\frac{V_{out}(s)}{V_{in}(s)} = -\frac{g_{m1} \cdot R_{out} \cdot \left(1 - \frac{s}{z_1}\right)}{1 - \frac{s}{p_1}}, \quad (2.28)$$

$$p_1 = -\frac{1}{R_{out} \cdot (C_{out} + C_M)} \approx -\frac{g_{m2}}{(C_{out} + C_M)}, \quad (2.29)$$

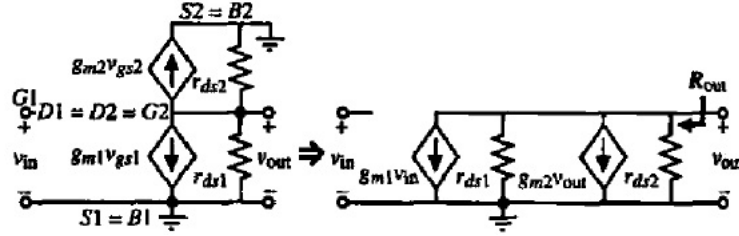


Figura 2.11: Modelo de pequenos sinais para um amplificador inversor com carga ativa.

2.5.1.2 Fonte de corrente

Assim como na configuração CMOS mostrada anteriormente, um transistor tipo N é usado como fonte comum. Em relação ao amplificador inversor com carga ativa, essa topologia apresenta uma excursão do sinal de saída maior, como pode ser visto na figura 2.12. Como o transistor M2 está polarizado com uma determinada tensão na porta, este transistor opera tanto no triodo como na saturação, dependendo da tensão de saída V_{out} . A curva de transferência estática pode ser encontrada, então, a partir da interseção entre os pontos das curvas de saída dos dois transistores, como mostrado na figura 2.12. Como o transistor correspondente a fonte de corrente não entra em corte até que a tensão entre seus terminais $V_{ds,2}$ se torne nula, a tensão máxima de saída $V_{out,max}$ atinge V_{DD} quando V_{in} se aproxima de V_{Th} , a tensão limiar do transistor N. A tensão mínima de saída $V_{out,min}$ desta topologia também é menor do que aquela do amplificador inversor com carga ativa, contribuindo também para a maior excursão do sinal de saída desta topologia.

O modelo de pequenos sinais deste amplificador pode ser encontrado ao igualar a corrente $I_{ds,2} = g_{m2} \cdot V_{out}$ no modelo anterior, na figura 2.11, a zero (o transistor correspondente a fonte de corrente possui um terra AC na porta). Com isso, a resistência de saída aumenta e consequentemente o ganho também aumenta, como pode ser verificado nas equações (2.31) e (2.30), respectivamente. Assim como na topologia com carga ativa, a função de transferência na equação (2.28) pode ser usada para determinar o ganho no domínio da frequência. No entanto, para o

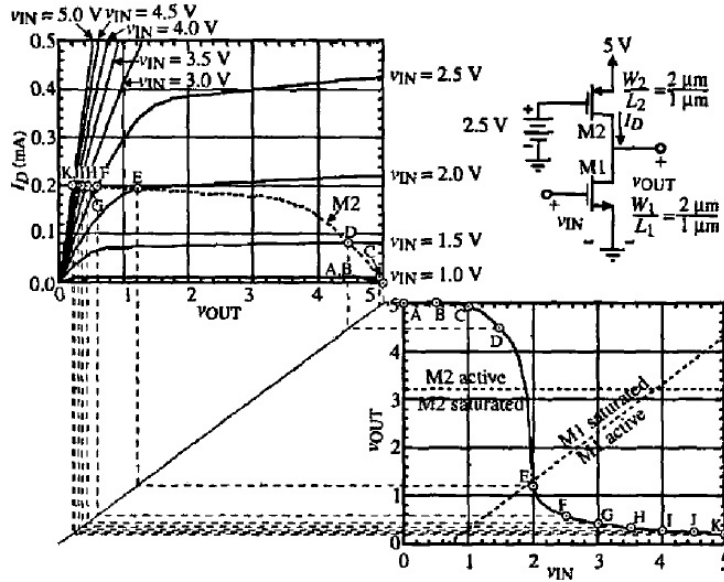


Figura 2.12: Curva de transferência de tensão de um amplificador inversor com fonte de corrente.

amplificador com fonte de corrente, devido a resistência de saída maior, o polo correspondente a frequência de -3 dB se torna menor na equação (2.32), reduzindo a banda do sinal de saída. Como esperado, o produto ganho-banda é mantido constante.

$$\frac{V_{out}}{V_{in}} = -\frac{g_{m1}}{g_{ds1} + g_{ds2}}, \quad (2.30)$$

$$R_{out} = \frac{1}{g_{ds1} + g_{ds2}}, \quad (2.31)$$

$$p_1 = -\frac{1}{R_{out} \cdot (C_{out} + C_M)} \approx -\frac{(g_{ds1} + g_{ds2})}{(C_{out} + C_M)}, \quad (2.32)$$

2.5.1.3 Push-pull

A curva de carga para esta topologia é obtida a partir das diversas curvas de saída do transistor tipo P, já que a tensão aplicada na porta de ambos transistores é variável e igual à V_{in} , como visto na figura 2.13. Isso implica em uma excursão do sinal de saída que varia de zero à V_{DD} , já que nenhum dos dois transistores corta até que $V_{ds,1}$ ou $V_{ds,2}$ se anule, permitindo que a tensão V_{out} se iguale a zero, através de M1 ou V_{DD} , através de M2.

A impedância de saída dessa topologia é igual a do amplificador com fonte de corrente, mostrada na equação (2.31). No entanto, o ganho, mostrado na equação (2.33) é maior, devido a contribuição de $I_{ds,2} = g_{m2} \cdot V_{in}$ na corrente de saída, como pode ser observado no modelo de pequenos sinais na figura 2.14. A banda dessa topologia é igual aquela do amplificador com fonte de corrente, pois a localização do polo e consequentemente a frequência de -3dB são mantidas as mesmas.

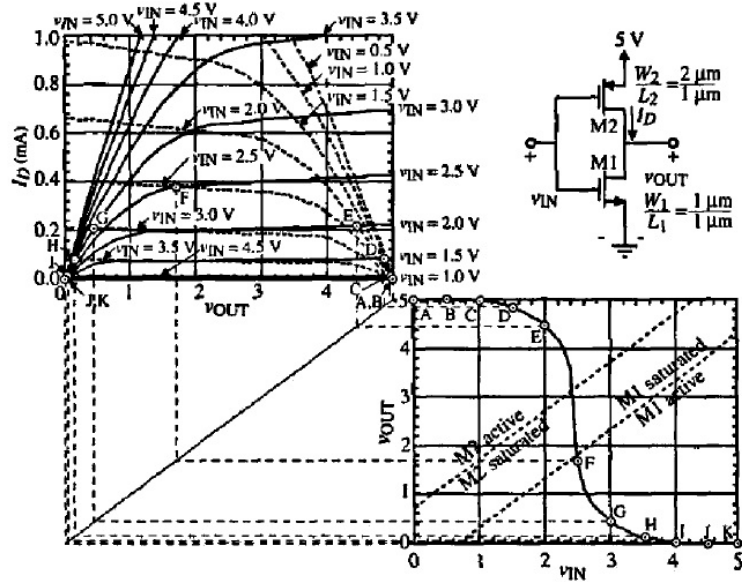


Figura 2.13: Curva de transferência de tensão de um amplificador inversor Push-pull.

$$\frac{V_{out}}{V_{in}} = -\frac{g_{m1} + g_{m2}}{g_{ds1} + g_{ds2}}, \quad (2.33)$$

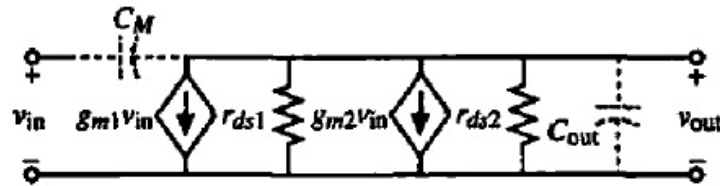


Figura 2.14: Modelo de pequenos sinais para um amplificador inversor Push-pull.

2.6 Verilog A e QUCS

Verilog A é uma linguagem de programação elaborada para implementar modelos de circuito analógicos. É uma das principais linguagens utilizadas para representação de modelos compactos, já que possui um nível de abstração alto e sua utilização não demanda do programador a solução de equações diferenciais e métodos numéricos para determinação das correntes.

Como já possui módulos correspondentes a estrutura de componentes ativos e passivos, a simples definição das equações e parâmetros de cada tecnologia é suficiente para o desenvolvimento do código. Muitos simuladores possuem compatibilidade com o Verilog A, assim como o QUCS, que é o simulador utilizado neste projeto.

O QUCS (*Quite Universal Circuit Simulator*) é um simulador gratuito de circuitos integrados, inicialmente desenvolvido para a plataforma LINUX. Além de simular modelos a partir de um código em Verilog A, é um simulador que também permite desenvolver componentes eletrônicos através da definição das equações correspondentes em uma interface gráfica, permitindo ao usuário

a verificação rápida das alterações em um modelo. Essa ferramenta é conhecida como *Equation defined Models* (Modelos definidos por equações). No QUCS, é possível fazer análises AC, DC, simular transientes, efeitos de ruídos, parâmetros S, entre outros.

Capítulo 3

Desenvolvimento

3.1 Introdução

Assim como introduzido no capítulo 1, esse projeto é a extensão do trabalho iniciado em [1], que consistiu em avaliar a implementação do modelo UMEM no simulador de circuitos QUCS para diversos OTFT's. No entanto, a implementação do modelo disponível no código em Verilog A usado no QUCS não foi compatível com o código criado no instituto desenvolvedor do UMEM, em uma outra plataforma (MATHCAD). Com isso, não foi possível utilizar os parâmetros do UMEM extraídos dessa plataforma (MATHCAD). Então, um conjunto padrão dos parâmetros do UMEM disponibilizados pelo código em Verilog A foi modificado com o propósito de ajustar as curvas de saída do QUCS às experimentais dos OTFT's.

Assim, para este trabalho, com o objetivo de simular um VOTFT no QUCS com os parâmetros extraídos pelo UMEM, modificou-se o código em Verilog A. No entanto, antes da alteração ser feita, o código no MATHCAD foi substituído por um equivalente no MATLAB, com o propósito de tornar a implementação do UMEM mais acessível, já que o MATLAB é uma plataforma muito mais difundida no meio acadêmico. Em seguida, otimizou-se a extração e o modelo UMEM no MATLAB através dos parâmetros de ajuste fornecidos pelo UMEM. Finalmente, comparou-se as implementações do modelo no MATHCAD e em Verilog A com o propósito de validar alteração do código e simular um VOTFT em topologias de circuito, as quais foram avaliadas na ultima etapa do projeto. O resumo do projeto é apresentado no fluxograma da figura 3.1. Cada etapa será detalhada nas seções posteriores.

3.2 Substituição do código no MATHCAD por um equivalente no MATLAB

Durante esse estágio do projeto, modificações foram feitas em um código com a implementação do UMEM no MATLAB fornecido pela Technische Universität Dresden (TUD), Alemanha, com o objetivo de usá-lo para substituir a implementação do UMEM no MATHCAD, tornando-o mais

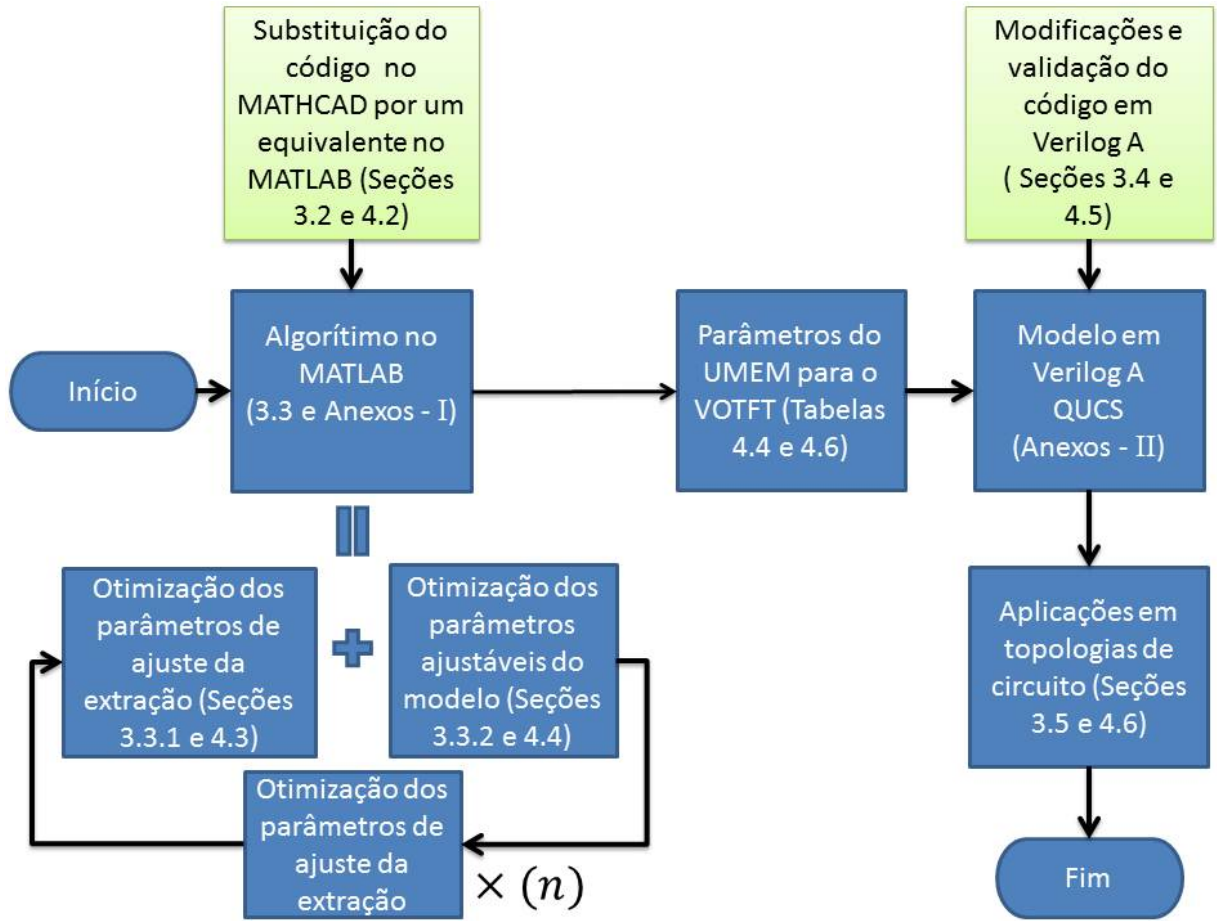


Figura 3.1: Fluxograma do projeto. Blocos em azul contém o fluxo para simulações de quaisquer OTFTs. Blocos em verde contém etapas realizadas somente para este projeto. A realimentação é feita apenas uma vez. A constante n indica um número suficiente de vezes que o bloco deve ser repetido antes do caminho direto ser executado novamente.

acessível e assim permitindo futuras otimizações e adições de novos módulos (como o modelo para capacitâncias parasitas e contatos não ôhmicos, por exemplo). Como o código fornecido pela TUD tinha sido criado para utilização em um conjunto de dados experimentais coletados de um TFT específico, a maioria das alterações feitas neste estágio inicial focaram em adaptar o código às diferentes condições de polarização do OTFT utilizado pelo CINVESTAV (instituto desenvolvedor do UMEM) no código no MATHCAD, a fim de verificar se os mesmos resultados eram obtidos da extração de parâmetros e aplicação do modelo nas duas plataformas (MATLAB e MATHCAD).

Em relação à extração, a primeira das modificações foi a alteração do tamanho dos vetores que armazenam os dados experimentais das curvas de saída $I_{ds} \times V_{ds}$ e transferência $I_{ds} \times V_{gs}$. Ou seja, como o OTFT utilizado no código implementado no MATHCAD (simplificado por CINVESTAV OTFT nas próximas citações) foi submetido à um *DCsweep* entre 0V e -30V, os vetores tiveram seus tamanhos alterados para 31 ou 301 pontos (o ultimo caso envolve o aumento em dez vezes do número de pontos das curvas através de interpolação numérica precedendo a extração). Outra alteração importante foram os valores de V_{gs} utilizados no cálculo das funções de ajuste.

Os últimos correspondem aos intervalos de $H(V_{gs})$ e $[I_{ds}(V_{gs})]^{\frac{1}{\gamma_a+2}}$ utilizados para extração dos parâmetros do modelo. Esse ajuste é necessário pois não é possível determinar valores para $H(V_{gs})$ e $[I_{ds}(V_{gs})]^{\frac{1}{\gamma_a+2}}$ que os tornem perfeitamente lineares ao longo de toda a curva. Assim, outras duas funções semelhantes foram definidas no código em MATHCAD: As retas $IPC(V_{gs})$ e $HH1(V_{gs})$, que aproximam as funções $[I_{ds}(V_{gs})]^{\frac{1}{\gamma_a+2}}$ (identificada por $IKS1(V_{gs})$ nas próximas citações) e $H(V_{gs})$, respectivamente, em um intervalo linear. Enquanto $HH1(V_{gs})$ é uma reta com coeficientes linear e angular da região em $H(V_{gs})$ escolhida para extração, $IPC(V_{gs})$ é encontrada através do produto entre o coeficiente angular de $IKS1(V_{gs})$, na região escolhida, e $(V_{gs} - V_T)$. Ou seja, a região na qual $IKS1(V_{gs})$ e $IPC(V_{gs})$ estão mais próximos é a região na qual pode-se estimar a inclinação em que a função $[I_{ds}(V_{gs})]^{\frac{1}{\gamma_a+2}}$ é linear, determinando consequentemente o parâmetro α_s . Assim, os intervalos V_{gs} usados para essas funções foram alterados para os valores considerados no MATHCAD como os melhores. Os demais parâmetros de ajuste das curvas do modelo Q, QS, DV, DVS, S e SS foram também modificados de acordo com o código no MATHCAD.

Outra alteração importante foi a do cálculo dos coeficientes das retas $HH1(V_{gs})$ e $IPC(V_{gs})$. Enquanto que para $HH1(V_{gs})$ a inclinação era calculada através da reta secante à $H(V_{gs})$ no intervalo escolhido, $IPC(V_{gs})$ utilizava a inclinação da reta tangente aos últimos dois pontos de $IKS1(V_{gs})$, como pode ser visto nas figuras 3.2 e 3.3. Por isso, utilizou-se a função *Polyfit* do MATLAB, a qual resulta nesses dois coeficientes calculados através de uma aproximação linear de toda a curva no intervalo V_{gs} escolhido.

```

61      % 3.2 Threshold voltage Vt
62
63 -    HH1                                = zeros(1,301);
64 -    Coefficients_Vt                    = polyfit(k1,l1,1);
65
66 -    P_Vt                               = Coefficients_Vt(1);
67 -    inter_Vt                           = Coefficients_Vt(2);
68
69      %P                                  = (l1(a+1)-l1(1))/(k1(a+1)-k1(1));
70      %inter                              = l1(1) - P*(k1(1));
71
72 -    for n2 = 1:301 %1:501
73 -        HH1(n2) = inter_Vt + P_Vt*(n2-1)/10;
74 -    end
75
76 -    Vt                                  = -(inter_Vt)/P_Vt;

```

Figura 3.2: Trecho do código no MATLAB antes (parte comentada) e após alteração da forma de calcular os parâmetros da reta $HH1(V_{gs})$.

Em relação a implementação do modelo, reduziu-se o número de vetores correspondentes às curvas de saída (número menor de tensões aplicadas na porta), assim como o tamanho dos vetores (tensão máxima V_{ds} reduzida de -50V no código original para -30V no CINVESTAV OTFT). Outra alteração importante foi a localização do ponto $V_{gs} = V_T + DV$ nos vetores correspondentes às curvas $I_s = I_{ds}(V_T + DV, V_{ds}) \cdot e^{\frac{2.3}{S} \cdot (V_{gs} - V_T)}$. No código fornecido pela TUD, esse ponto foi

```

151 - for n4 = 1:301
152 -     iiks1(n4) = mpower(iksatcor_p(n4), 1/(2+gamma));
153 - end
154
155 - for i1 = 1:aS+1
156 -     ips1(i1) = iiks1(100+i1);
157 - end
158
159 - Coefficients_alphas = polyfit(ks1,ips1,1);
160
161 - P_alphas = Coefficients_alphas(1);
162 - % P_alphas = (ips1(aS+1)-ips1(aS))/(ks1(aS+1)-ks1(aS));
163
164 - Ps = P_alphas;
165
166
167 - for x = 1:251
168 -     ipc(x) = Ps*((x-1)*0.2)-Vt);
169 - end
170
171 - alphas = ((Vaa^gamma)/K)*(Ps^(2+gamma))*sqrt(2);

```

Figura 3.3: Trecho do código no MATLAB antes (parte comentada) e após alteração da forma de calcular o coeficiente angular da reta $IPC(V_{gs})$.

encontrado com base no conhecimento prévio do parâmetro extraído V_T e uma interpolação de I_{ds} foi feita de forma a considerar todas as casas decimais de V_T como uma posição dentro do vetor corrente. Isso foi feito por causa da modulação do termo $e^{\frac{2.3}{S} \cdot (V_{gs} - V_T)}$, que traz erros consideráveis na determinação de I_s caso casas decimais de V_T sejam desprezadas. No código modificado, o número de pontos do vetor original foi aumentado 1000 vezes, com o objetivo de considerar as 3 primeiras casas decimais da tensão V_T para quaisquer transistores usados no UMEM. Na figura 3.4 mostra-se o trecho do código utilizado para localizar V_T no vetor corrente, considerando o aumento do número de pontos.

```

422 - for Vgs = -5:50
423 -     ISc(Vgs+6) = idc_Vds5_inter(ceil((Vt+DV)*1000)+1)*(exp(((Vgs-Vt)/S)*2.3));
424 -     % ISc(Vgs+1) = idc_Vds5_inter(82)*(exp(((Vgs-Vt_Vg1)/S)*2.3));
425 - end

```

Figura 3.4: Trecho do código no MATLAB mostrando como V_T é localizado no vetor corrente I_{ds} com a consideração de três casas decimais (multiplicação de V_T por 1000).

Após essas modificações, comparou-se as implementações do UMEM nos dois códigos (MATLAB modificado e MATHCAD) através das curvas modeladas e dos parâmetros extraídos do CINVES-TAV OTFT. Nota-se que o ajuste satisfatório entre as curvas não é suficiente. Assim como foi mostrado em [1], diferentes parâmetros extraídos podem convergir nas mesmas curvas (ou torná-las muito próximas). Logo, espera-se também que os parâmetros encontrados sejam suficientemente próximos.

3.3 Aprimoramento do código no MATLAB

Principal objetivo desta seção é minimizar o número de alterações necessárias no código em MATLAB a cada novo OTFT modelado pelo UMEM, tornando-o mais genérico. O código fornecido é composto de quatro módulos, sendo o primeiro deles responsável principalmente pela definição dos parâmetros físicos do transistor, tais como largura, comprimento do canal, constante relativa do dielétrico utilizado, espessura do isolante, além de tensões máximas medidas para as curvas de transferência e saída do transistor. O segundo módulo é responsável pelo carregamento no MATLAB dos dados experimentais correspondentes as curvas de transferência e saída medidas, as quais devem estar inicialmente armazenadas em um arquivo com extensão .xls(excel). O terceiro módulo contém o código que realiza a interpolação dos dados experimentais, já que o número de pontos coletados para algumas curvas pode ser insuficiente quando utilizadas na extração de parâmetros. A interpolação é sempre feita através de funções cúbicas, as quais fornecem uma representação satisfatória das curvas $I_{ds}xV_{ds}$ e $I_{ds}xV_{gs}$. O ultimo módulo corresponde a própria implementação do UMEM, a qual inclui extração de parâmetros e o modelo.

Uma das alterações feitas foi dar ao usuário o controle do número de pontos usados na interpolação dos dados experimentais durante a extração. Ou seja, o tamanho de todos os vetores de entrada e saída contendo as curvas experimentais e modeladas pelo UMEM, respectivamente, foram fixados em um valor N determinado pelo usuário. Para isso, o passo entre uma medida e outra foi modificado para $\frac{V_{max}}{N-1}$, onde V_{max} é o máximo valor medido de tensão das curvas de transferência ou saída. Com isso, o usuário pode verificar como os dados experimentais são interpolados a cada simulação e determinar um número mínimo de pontos que produza uma aproximação satisfatória, sem demandar muito esforço computacional. O número N desejado é adicionado ao primeiro módulo do código.

Outra alteração importante na generalização do código foi a exclusão da determinação de parâmetros (do modelo) através da localização de pontos nos vetores por números. Ao invés disso, funções do MATLAB como *length* foram utilizadas, tornando o código mais compreensível em relação como esses parâmetros estão sendo determinados. Por exemplo, na figura 3.5, observa-se que o conhecimento do número de medidas realizadas na determinação da curva de transferência torna-se dispensável, pois a função $H(V_{gs})$ é calculada, independentemente, para todos os valores de corrente.

Houve também uma alteração na forma de selecionar a corrente de saída apropriada para determinação dos parâmetros λ e m . Recomenda-se da literatura em [7] que se utilize a curva de saída correspondente ao máximo valor de V_{gs} que deseja-se modelar. Sabe-se que isso depende também do máximo valor V_{gs} para o qual a curva de transferência teve suas medidas realizadas, o qual influencia na determinação do parâmetro α_s e conseqüentemente $V_{ds,SAT}$. Ou seja, a curva de saída escolhida deve ser a maior possível e corresponder à um valor de V_{gs} medido na curva de transferência. Para satisfazer as duas condições, o código na figura 3.6 foi implementado. A intenção é encontrar um valor de V_{gs} da curva de saída imediatamente inferior ou igual ao seu valor máximo medido na curva de transferência.


```

59      % 3.1 Trapezoidal integration approach to calculate H1
60
61
62
63 -   for n2 = 1:length(idllkcor_p) % 1:501
64 -       cal      = cumtrapz(n1,idllkcor_p);
65 -       H1(1,n2) = cal(n2)/idllkcor_p(n2);
66 -   end

```

Figura 3.5: Trecho do código no MATLAB mostrando como $H(V_{gs})$ é determinado, através da localização de pontos em um vetor com valores das integrais calculadas para todas as medidas de corrente. O número comentado mostra qual era o tamanho do vetor após interpolação, informação que não precisa mais ser inserida.

```

160      % 4. Current to calculate parameters: m and lambda
161
162 -   xx_interp_measured_cor_p = 0:0.001:Vdmax;
163 -   Vector_positions        = find(abs(min(abs(VgsmaxS - Vgs_out)) -...
164 -       abs(VgsmaxS - Vgs_out)) < 1e-10);
165 -   Vector_Position        = min(Vector_positions);
166 -   imaxcor_p              = interp1(xx_spline_measured_cor,...
167 -       icor(Vector_Position,:), xx_interp_measured_cor_p,'linear');
168 -   imaxcor                 = icor(Vector_Position,:);

```

Figura 3.6: Trecho do código no MATLAB mostrando como a curva de saída apropriada, *imaxcor*, é escolhida para determinação dos parâmetros λ e m .

Em relação aos gráficos, foram incluídos, além dos correspondentes às curvas de transferência e saída (essenciais na análise do UMEM), outros que ajudam na determinação das regiões das funções de ajuste $H(V_{gs})$ e $I_{KS1}(V_{gs})$ usadas para extração de parâmetros do UMEM, como será mostrado na próxima seção. Com o objetivo de fornecer maiores recursos durante a análise dos resultados, códigos correspondentes ao cálculo e plotagem das transcondutâncias experimentais e modeladas pelo UMEM foram também incluídos.

Em suma, poucas alterações precisam ser feitas no código para aplicação do UMEM a outros OTFT's. Somente os módulos correspondentes à definição de parâmetros do transistor e ao carregamento de dados experimentais no MATLAB precisam ser modificados. Como a quantidade de medidas feitas é algo que depende do experimento, tal informação afeta o tamanho dos vetores e a quantidade (número de curvas de saída). Além disso, como os dados são carregados a partir de uma planilha no excel, a linha correspondente à ultima leitura precisa ser alterada no código no MATLAB. Em relação ao módulo correspondente à definição dos parâmetros do OTFT, deve-se mudar principalmente aqueles correspondentes às dimensões do transistor, constante relativa do dielétrico, além das tensões máximas medidas para cada curva. Foi também incluído nesse módulo o controle da região para a qual as funções de ajuste da extração são calculadas, procedimento que será detalhado na próxima seção.

3.3.1 Otimização dos parâmetros de ajuste da extração

Como mencionado anteriormente, essa parte consiste em descrever como um intervalo apropriado de V_{gs} para o cálculo das funções de ajuste pode ser determinado. Inicialmente, define-se como parâmetros de ajuste da extração os limites mínimo e máximo do intervalo V_{gs} para o qual $H(V_{gs})$ e $[I_{ds}(V_{gs})]^{\frac{1}{\gamma_a+2}}$ (identificado por $IKS1(V_{gs})$ ao longo do texto) são aproximados por retas. Os intervalos utilizados para as duas funções não são necessariamente iguais e por isso possuem denominações diferentes. Para os limites inferior e superior do intervalo de $H(V_{gs})$, usa-se os termos $V_{H,min}$ e $V_{H,max}$, respectivamente. Analogamente, define-se como limites para o intervalo de $IKS1(V_{gs})$ os termos $V_{IPC,min}$ e $V_{IPC,max}$. O motivo de serem denominados como parâmetros de ajuste da extração baseia-se no fato de alterações em seus valores mudarem os parâmetros extraídos. Como a inclinação dessas curvas não é a mesma ao longo de todos valores de V_{gs} medidos, escolhe-se o intervalo para o qual a extração resulta na melhor aproximação do modelo UMEM com as curvas experimentais. Para isso, elaborou-se um simples algoritmo baseado nas recomendações dos desenvolvedores do UMEM em [7].

No código em MATHCAD, recomenda-se a escolha de uma região linear da curva $H(V_{gs})$ cujo o limite superior seja menor que o máximo valor medido de V_{gs} na curva de transferência. Assim, com o propósito de convergir nessa solução, inicia-se o primeiro passo do algoritmo por um intervalo mais próximo possível do valor máximo (preferencialmente incluindo este ponto). Como não se conhece previamente o gráfico da função $H(V_{gs})$ antes da primeira simulação, usa-se um intervalo com comprimento qualquer. No entanto, recomenda-se a escolha de um intervalo com comprimento igual a $\frac{1}{3}$ da máxima tensão medida, a fim de evitar muitas simulações, já que a não-linearidade da curva $H(V_{gs})$ não é tão acentuada para a maioria dos valores de V_{gs} medidos. Ou seja, é escolhido um intervalo entre $\frac{2}{3} \cdot V_{gs,max}$ e $V_{gs,max}$ para a primeira simulação.

Em relação à função $IKS1(V_{gs})$, escolhe-se para o passo inicial o mesmo intervalo escolhido para função $H(V_{gs})$, já que geralmente os melhores intervalos para as duas funções estão próximos à tensão máxima medida $V_{gs,max}$ da curva de transferência.

Para o segundo passo do algoritmo, observa-se, inicialmente, se a escolha do intervalo para a função $IKS1(V_{gs})$ é razoável. As maiores discrepâncias entre o modelo do UMEM e as curvas experimentais são notadas quando $IKS1(V_{gs})$ se afasta muito de $IPC(V_{gs})$. Como mencionado, essas duas curvas são utilizadas somente para estimar o coeficiente angular da função $[I_{ds}(V_{gs})]^{\frac{1}{\gamma_a+2}}$. Se a distância entre elas é considerável no intervalo escolhido, a inclinação será encontrada para uma região não linear da função $IKS1(V_{gs})$ e a extração de parâmetros do UMEM produzirá resultados inadequados. Caso o intervalo para $IKS1(V_{gs})$ precise ser alterado, o intervalo sugerido é aquele que mais aproxime as funções $IPC(V_{gs})$ e $IKS1(V_{gs})$. Assim, mantém-se constante o intervalo correspondente a função $H(V_{gs})$ e executa-se uma nova simulação.

Para o terceiro passo do algoritmo, mantém-se constante os parâmetros de ajuste de $IKS1(V_{gs})$ e desloca-se o intervalo da função $H(V_{gs})$ em direção a menores valores de V_{gs} , até que o ajuste entre as curvas modeladas e experimentais seja ótimo, o que não ocorre necessariamente no intervalo de maior linearidade de $H(V_{gs})$. Há alguns critérios que devem ser seguidos durante essa etapa. Caso $H(V_{gs})$ seja uma curva que se afaste muito de uma reta, reduz-se o tamanho dos intervalos

usados ou o passo entre eles, tal que se obtenha uma variação menor do ajuste entre as curvas e os parâmetros extraídos a cada simulação. Com isso, é possível obter uma solução mais precisa.

Nota-se que nem sempre o segundo passo do algoritmo corresponde a uma simulação, pois a escolha do intervalo para a função $\text{IICS1}(V_{\text{gs}})$ no primeiro passo do algoritmo pode ter sido satisfatória. Isso é o que geralmente ocorre, reduzindo o número de simulações necessárias para o UMEM.

O algoritmo acima foi aplicado a dois transistores: Um OTFT fornecido pela Technische Universität Dresden (simplificado por TUD OTFT nas próximas citações) e um TFT orgânico com arquitetura vertical (simplificado por TUD VOTFT) obtido da literatura em [12]. Ambos são tipo P e informações relativas a alguns parâmetros físicos fornecidos, além de outros extraídos pelo UMEM, podem ser encontrados na tabela 3.3 no final deste capítulo.

Em relação ao TUD OTFT, o algoritmo foi aplicado duas vezes: Primeiramente, para um intervalo V_{gs} com tamanho igual a 2V, para o qual três simulações foram feitas. Em seguida, o algoritmo foi aplicado para um intervalo com tamanho igual a 1V, para o qual quatro simulações foram realizadas.

Em relação ao intervalo com tamanho igual a 2V, os seguintes procedimentos foram adotados: Para primeira simulação, escolheu-se intervalos iguais para as duas funções $H(V_{\text{gs}})$ e $\text{IICS1}(V_{\text{gs}})$, cujo limite superior foi igual ao valor máximo da tensão medida na curva de transferência $V_{\text{gs,max}}$. A partir da visualização das duas curvas de ajuste, observou-se que o intervalo escolhido era o que resultava em maior aproximação entre as funções $\text{IICS1}(V_{\text{gs}})$ e $\text{IPC}(V_{\text{gs}})$. Ou seja, os parâmetros de ajuste $V_{\text{IPC,min}}$ e $V_{\text{IPC,max}}$ não foram alterados nas demais simulações. Notou-se no outro gráfico gerado que a função $\text{HH1}(V_{\text{gs}})$, usada para aproximar $H(V_{\text{gs}})$ na região escolhida por uma reta, não se afastava de $H(V_{\text{gs}})$ dentro deste intervalo, confirmando que $H(V_{\text{gs}})$ possuía uma aproximação linear satisfatória. No entanto, como isso é uma condição necessária, mas não suficiente, reduziu-se os valores dos parâmetros de ajuste $V_{\text{H,min}}$ e $V_{\text{H,max}}$ nas duas próximas simulações até que o ajuste às curvas experimentais para o TUD OTFT fosse o melhor possível.

Em relação ao intervalo com tamanho igual a 1V, os mesmos procedimentos descritos acima foram adotados. No entanto, o número de simulações foi maior, por causa redução do tamanho do intervalo. A intenção era verificar que a redução do tamanho do intervalo para uma função $H(V_{\text{gs}})$ aproximadamente linear não produz muitas diferenças na extração de parâmetros do modelo ou na aproximação com as curvas experimentais. A validação disso pode ser verificada no capítulo 4.

Para o TUD VOTFT, o algoritmo foi aplicado somente uma vez, para o qual 3 simulações foram realizadas. De forma semelhante ao TUD OTFT, o intervalo inicial usado estava entre $\frac{2}{3} \cdot V_{\text{gs,max}}$ e $V_{\text{gs,max}}$ para as duas funções $\text{IICS1}(V_{\text{gs}})$ e $H(V_{\text{gs}})$. No entanto, após a primeira simulação, notou-se maiores não-linearidades de $H(V_{\text{gs}})$, o que levou a redução do passo V_{gs} entre os intervalos para as próximas simulações. Foi observado também que o intervalo escolhido não correspondia ao de distância mínima entre as funções $\text{IICS1}(V_{\text{gs}})$ e $\text{IPC}(V_{\text{gs}})$. Ou seja, o segundo passo do algoritmo foi aplicado e durante a segunda simulação, os parâmetros de ajuste $V_{\text{H,min}}$ e $V_{\text{H,max}}$ foram mantidos constantes. Para a terceira simulação, após a escolha do melhor intervalo para $\text{IICS1}(V_{\text{gs}})$, os parâmetros de ajuste dessa função foram mantidos constantes, ao passo que o

intervalo da função $H(V_{gs})$ foi deslocado novamente, com o propósito de encontrar o melhor ajuste possível às curvas experimentais.

Os parâmetros de ajuste da extração usados para o TUD VOTFT e TUD OTFT (para um intervalo de tamanho igual à $2V$) nas três simulações podem ser encontrados na tabela 3.1.

Tabela 3.1: Parâmetros de ajuste da extração para o TUD VOTFT (parte superior da tabela) e TUD OTFT (parte inferior da tabela) para um intervalo com tamanho igual a $2V$. Os dados da tabela correspondem às três simulações.

Parâmetro de ajuste - TUD VOTFT	1ª simulação	2ª simulação	3ª simulação
$V_{H,min}$ [V]	-4	-4	-3.5
$V_{H,max}$ [V]	-6	-6	-5.5
$V_{IPC,min}$ [V]	-3	-1.8	-1.8
$V_{IPC,max}$ [V]	-5	-2.2	-2.2
Parâmetro de ajuste - TUD OTFT	1ª simulação	2ª simulação	3ª simulação
$V_{H,min}$ [V]	-4	-3	-2
$V_{H,max}$ [V]	-6	-5	-4
$V_{IPC,min}$ [V]	-4	-4	-4
$V_{IPC,max}$ [V]	-6	-6	-6

3.3.2 Otimização dos parâmetros ajustáveis do modelo

Essa seção apresenta a inclusão de uma nova parte no código ainda não mencionada, cujo propósito é reduzir o número de parâmetros ajustáveis fornecidos pelo UMEM. Diferentemente da seção anterior, estes não afetam a extração de parâmetros do modelo (V_T , γ_a , V_{aa} , R , α_s , m e λ), mas somente o ajuste das curvas de saída e transferência, assim como explicado no capítulo 2. Os parâmetros ajustáveis do modelo abordados nesta seção são, então, os parâmetros Q , QS , DV , DVS , S e SS já mencionados anteriormente.

Algumas formas de determinar estes parâmetros foi sugerido em [9]. No entanto, tais métodos dependem da visualização de regiões nas curvas, o que pode torná-los imprecisos. É sugerido, por exemplo determinar os parâmetros S e SS através das inclinações das curvas de transferências na região sublimiar em gráficos com escala logarítmica. Tentou-se, assim, encontrar esses parâmetros para o CINESTAV OTFT. No entanto, os valores obtidos variavam muito e estavam distantes daqueles usados no código no MATHCAD. Outra sugestão feita em [9] foi a determinação dos parâmetros DV e DVS através da localização do valor de tensão para o qual as correntes cresciam exponencialmente na região sublimiar das curvas de transferência. Para os parâmetros Q e QS , não foram feitas sugestões.

Como a finalidade de todos esses parâmetros Q , QS , DV , DVS , S e SS é ajustar as curvas do modelo, é proposto um método simples de obtê-los, o qual consiste em minimizar o erro relativo das curvas modeladas pelo UMEM em relação às curvas experimentais, determinado pela equação: $Erro(\%) = 100 \cdot \left(\frac{I_{ds,Exp} - I_{ds,UMEM}}{I_{ds,Exp}} \right)$. O erro é, então, calculado para as diversas variações

das correntes modeladas pelo UMEM encontradas através de combinações dos parâmetros acima. Em seguida, os erros correspondente a cada associação entre Q, DV e S são armazenados em um vetor.

No entanto, nota-se que o erro é calculado individualmente para cada ponto dos vetores $I_{ds,Exp}$ e $I_{ds,UMEM}$. Logo, armazena-se no vetor mencionado acima as médias e os valores máximos dos erros encontrados para cada curva $I_{ds,UMEM}$ modelada. Assim, define-se duas formas de minimizá-lo: Através da determinação da menor média ou menor valor máximo desses valores armazenados. A curva otimizada, correspondente ao menor erro é, então, obtida com o uso dos dois métodos. Todo o procedimento é realizado automaticamente através de uma parte nova do código implementada. Para essa seção do projeto, usou-se os transistores TUD OTFT e TUD VOTFT.

Outra observação importante é o uso de somente uma das curvas de transferências para determinação do menor erro. Isso é tido como suficiente, pois o uso de qualquer uma das curvas de transferência produz resultados gráficos semelhantes. As curvas de saída, no entanto, não são apropriadas para determinação de todos esses parâmetros, pois S e SS somente se aplicam as curvas de transferência, a fim de ajustá-las aos dados experimentais na região sublimiar.

É importante mencionar que os valores dos parâmetros encontrados manualmente (várias simulações executadas até encontrar os parâmetros que resultassem no melhor ajuste entre as curvas) foram utilizados como referência para comparação com os parâmetros encontrados com a otimização do código. Para ambos transistores, TUD OTFT e TUD VOTFT, esses parâmetros de referência foram usados na seção anterior (seção da otimização dos parâmetros de ajuste da extração).

Para fins de simplificação, a tabela 3.2 associa os diversos parâmetros mencionados neste projeto com suas denominações.

Tabela 3.2: Conjunto de todos os parâmetros mencionados neste projeto. Resumidamente: Parâmetros de ajuste da extração, parâmetros ajustáveis do modelo e parâmetros extraídos do modelo.

Ajuste da extração	Ajustáveis do modelo	Extraídos do modelo
$V_{H,min}$	Q	V_T, R
$V_{H,max}$	QS	γ_a, V_{aa}
$V_{IPC,min}$	DV, DVS	m, λ
$V_{IPC,max}$	S, SS	α_s

3.4 Modificações do código em Verilog A

Como mencionado no capítulo 1, o objetivo principal desta seção foi identificar a diferença entre a implementação do modelo do UMEM nas plataformas MATHCAD e QUCS. No entanto, com a validação da implementação do UMEM no MATLAB, o MATHCAD não foi mais usado. Assim, após as mudanças no código em Verilog A, as curvas de saída e transferência obtidas através do MATLAB e QUCS foram comparadas, com o proposito de validar o novo código em Verilog

A e o QUCS como um simulador de OTFT's, cujos parâmetros podem ser extraídos através da implementação do UMEM no MATLAB.

Um dos motivos da corrente encontrada em [1] ter sido aproximadamente 100 vezes maior no QUCS deve-se as unidade de medidas usadas na definição de parâmetros no código em Verilog A. Não somente os parâmetros físicos, como largura, comprimento do canal e espessura do isolante, mas também constantes como a permissividade no vácuo e constante dielétrica do isolante estavam dimensionadas em metros. Isso tornava a constante multiplicativa K da corrente 100 vezes maior, causando diferenças entre as curvas na mesma ordem de grandeza.

Outra diferença importante entre as duas implementações é que as correntes de saída modeladas no MATHCAD, assim como no MATLAB(a partir deste projeto), consideram, de acordo com a literatura do UMEM, efeitos da histerese nos transistores orgânicos. Ou seja, usa-se três equações para modelar a corrente de saída, dependendo das condições de polarização. Por exemplo, para a curva de saída, soma-se à I_{ds} um fator de correção $\frac{V_{ds}}{CC}$. Para a curva de transferência soma-se à I_{ds} uma constante correspondente a corrente de fuga, a qual muda de acordo com a tensão V_{ds} aplicada. No entanto, nota-se no código em Verilog A, o uso de apenas uma equação, pois não é possível aplicar em um simulador de circuitos diferentes equações que variem de acordo com a polarização de um dispositivo. Assim, outra alteração do código em Verilog A foi somar a corrente total I_{ds} ao fator de correção $\frac{V_{ds}}{CC}$ e considerar as correntes de fuga iguais a zero (antes somadas à I_{ds} em diferentes regiões de operação do transistor), para quaisquer condições de polarização. O objetivo é reduzir os fatores intrínsecos de cada experimento em simulações de circuitos, para os quais as tensões de polarização, excursões de sinal não são os mesmos. Espera-se, então, que caso o ajuste às curvas experimentais usadas na extração não seja consideravelmente afetado por essas considerações, possa-se aproximar a corrente de saída por uma equação que não seja tão dependente dos efeitos da histerese (correntes de fuga desprezadas).

A constante CC mencionada anteriormente pode ser aproximada como o inverso da inclinação da curva de saída para $V_{gs} = 0V$. Ou seja, espera-se que a alterações provocadas por CC nas curvas seja muito pequena, pois para maioria dos dispositivos a tensão limiar V_T é positiva e $I_{ds} = 0V$ para $V_{gs} = 0V$.

Assim verifica-se para as três tecnologias diferentes usadas neste trabalho (tabela 3.3) se essas modificações são válidas e se o código modificado em Verilog A implementa o modelo do UMEM sem perdas consideráveis.

3.5 Aplicações de VOTFT em topologias de circuito

Após a validação do código em Verilog A, utiliza-se os parâmetros extraídos do UMEM no MATLAB para simular o TUD VOTFT em algumas topologias de circuito. Assim, é possível avaliar como o ajuste encontrado para cada uma das curvas afeta o comportamento esperado do circuito. Para essa seção foram feitas somente análises DC, para as quais curvas de transferência estática foram plotadas. As seguintes topologias foram simuladas: Amplificadores inversores fonte comum com carga resistiva, com carga ativa, fonte de corrente e push-pull. Para as ultimas três

topologias citadas, foi utilizada uma configuração CMOS. No entanto, como o TUD VOTFT é tipo P, usou-se os mesmos parâmetros do UMEM no transistor tipo N, com exceção de V_T , que teve sinal trocado. Na figura 3.7, pode-se ver a interface do QUCS com o usuário para inserção dos parâmetros do modelo em cada transistor. No caso do VOTFT simulado, foram utilizados os parâmetros obtidos através das otimizações mencionadas nas seções anteriores.

Tabela 3.3: Parâmetros genéricos para OTFT's, geralmente encontrados em outros modelos. Na tabela, R corresponde a resistências dos contatos ôhmicos e μ_{fet0} a mobilidade para campos elétricos baixos. Os parâmetros extraídos abaixo para o TUD OTFT e TUD VOTFT são resultados da otimização.

Parâmetros físicos	CINVESTAV OTFT	TUD OTFT	TUD VOTFT
Tipo do semiconductor	P	P	P
W [cm]	0.021	0.1	0.12
L [cm]	3×10^{-3}	0.005	5×10^{-6}
di [cm]	3.9×10^{-5}	3×10^{-6}	2.3×10^{-6}
Parâmetros extraídos	CINVESTAV OTFT	TUD OTFT	TUD VOTFT
V_T [V]	1	-1.19	-0.68
R [Ω]	3.15×10^6	6.65×10^3	3.46×10^3
λ [1/V]	-6.49×10^{-4}	-0.0212	-7.77×10^{-3}
μ_{fet0} [$\text{cm}^2 \text{V}^{-1} \text{s}^{-1}$]	8.18×10^{-4}	0.2838	3.1×10^{-4}

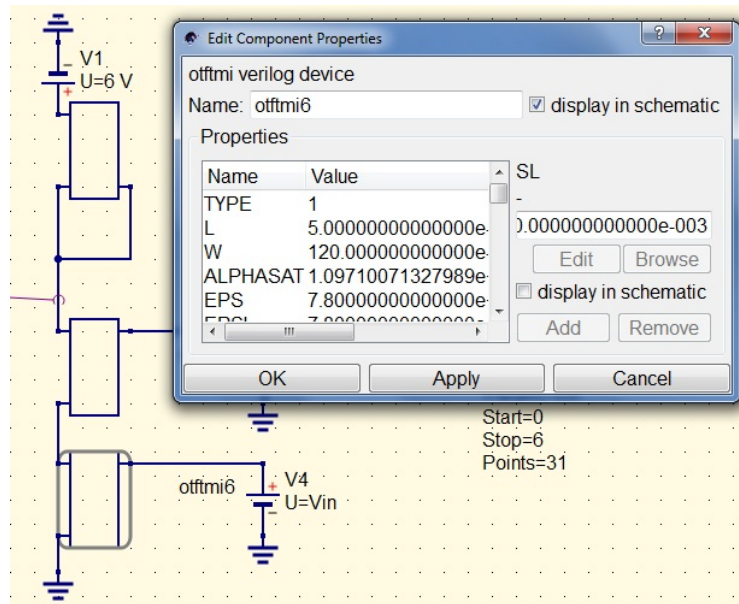


Figura 3.7: Interface do QUCS com o usuário para inserção de parâmetros do modelo em cada transistor.

Capítulo 4

Resultados

4.1 Introdução

Apresenta-se nesta seção os resultados correspondentes a todos procedimentos descritos no capítulo 3. Inicialmente, obtém-se a validação das alterações do código no MATLAB. Em seguida, verifica-se se as otimizações produzem melhores resultados. Finalmente, com a validação do código em Verilog A, utiliza-se os parâmetros de ajuste otimizados encontrados para o VOTFT, a fim de analisar se o seu desempenho em topologias de circuito corresponde ao esperado.

4.2 Validação da implementação do UMEM na plataforma MATLAB

Como descrito no capítulo anterior, os resultados das alterações feitas no código em MATLAB fornecido pela Technische Universität Dresden (Alemanha) podem ser comparados com a implementação do UMEM no MATHCAD, a fim de observar se as modificações foram adequadas. Os gráficos 4.1, 4.2 e 4.3 representam os resultados para o CINVESTAV OTFT, cuja extração de parâmetros e aplicação do modelo do UMEM foram feitas no MATHCAD e MATLAB.

Como pode ser observado, as duas plataformas fornecem os mesmos gráficos, comprovando que a implementação do UMEM nas duas é equivalente. No entanto, isso não é suficiente para afirmar que os parâmetros extraídos são os mesmos, já que parâmetros diferentes podem fornecer o mesmo ajuste entre as curvas. Por essa razão, as extrações feitas nas duas plataformas também foram comparadas e os valores apresentados na tabela 4.1 mostram que os parâmetros encontrados estão suficientemente próximos para afirmar que os dois códigos são equivalentes.

Nota-se que os valores das correntes obtidas experimentalmente nos gráficos 4.2 e 4.3 se afastam muito daquelas modeladas nas duas plataformas para $|V_{gs}| > 30V$. No entanto, isso não deve ser considerado uma limitação do UMEM, mas um erro obtido da extrapolação dos dados experimentais implementada no código em MATHCAD. Como o valor máximo de V_{gs} medido para essas curvas foi $-30V$, o crescimento resultante da extrapolação não pode ser considerado como real,

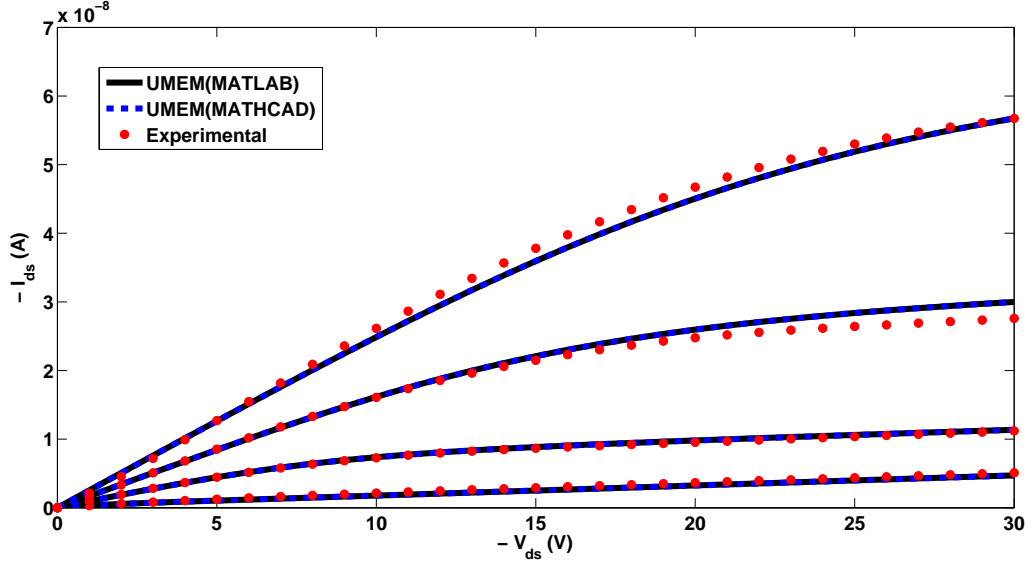


Figura 4.1: Curvas características de saída do CINVSTAV OTFT encontradas experimentalmente e modeladas pelo UMEM no MATLAB e MATHCAD, usando os parâmetros da tabela 4.1 para $V_{gs} = 0V, -10V, -20V$ e $-30V$.

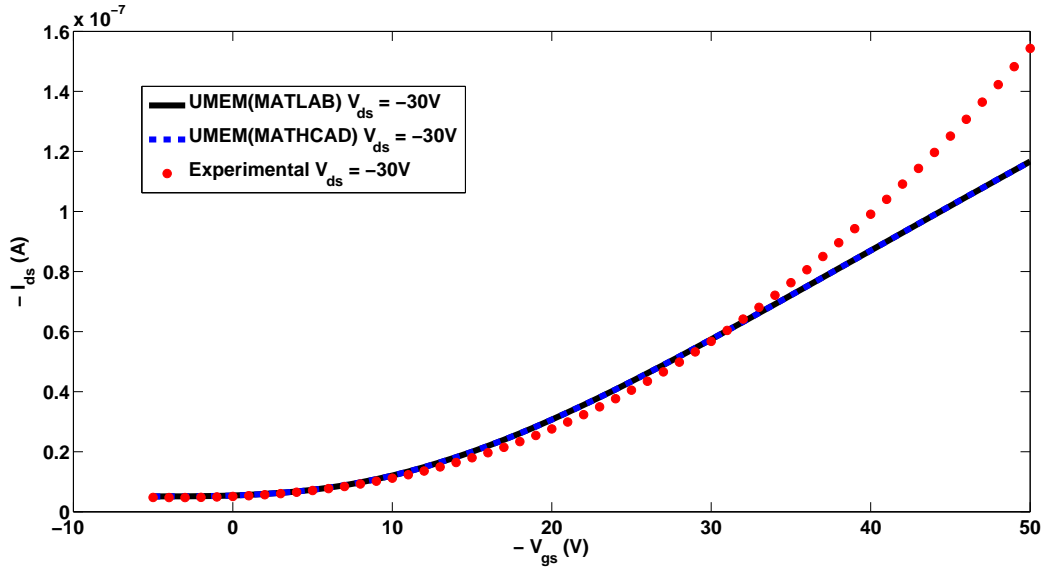


Figura 4.2: Curvas características de transferência do CINVSTAV OTFT encontradas experimentalmente e modeladas pelo UMEM no MATLAB e MATHCAD, usando os parâmetros da tabela 4.1

mas como somente uma aproximação da curva por uma função cúbica. Consequentemente, para os demais dispositivos modelados nesse trabalho, os dados experimentais não foram extrapolados.

A discrepância entre os valores de V_{aa} na tabela 4.1 é causada pelas diferenças na terceira e quarta casas decimais entre os parâmetros utilizados para seu cálculo. No entanto, V_{aa} é somente um parâmetro de ajuste utilizado para encontrar a mobilidade no canal, cujo valor é aproxi-

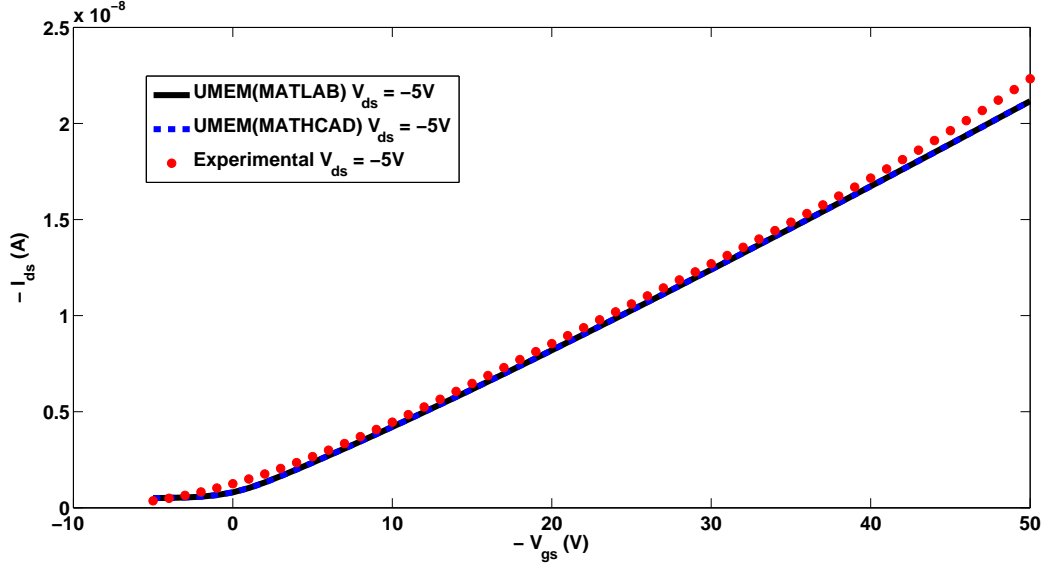


Figura 4.3: Curvas características de transferência do CINVESTAV OTFT encontradas experimentalmente e modeladas pelo UMEM no MATLAB e MATHCAD, usando os parâmetros da tabela 4.1

madamente o mesmo nas duas plataformas. Para o MATLAB, o valor encontrado foi $\mu_{\text{fet0}} = 8.18 \times 10^{-4} \text{ cm}^2 \text{ V}^{-1} \text{ s}^{-1}$, enquanto no MATHCAD, $\mu_{\text{fet0}} = 8.2 \times 10^{-4} \text{ cm}^2 \text{ V}^{-1} \text{ s}^{-1}$.

Conclui-se que a extração de parâmetros e aplicação do modelo do UMEM através dos códigos em MATHCAD e MATLAB produzem resultados equivalentes. Consequentemente, o uso do MATHCAD pode ser substituído pelo MATLAB, permitindo a elaboração de um código otimizado e mais genérico, o qual será avaliado na próxima seção. A tabela 4.1 contém os parâmetros do UMEM obtidos através do MATHCAD e MATLAB.

Tabela 4.1: Parâmetros do modelo UMEM extraídos através do MATHCAD e MATLAB para o CINVESTAV OTFT. Esses dados são utilizados para modelar as curvas 4.1, 4.2 e 4.3

Parâmetro	MATHCAD	MATLAB
V_T [V]	1	1
γ_a	0.117	0.117
V_{aa} [V]	3.073×10^{26}	2.497×10^{26}
R [Ω]	3.14×10^6	3.15×10^6
α_s	0.847	0.847
m	3.11	3.11
λ [1/V]	-6.4×10^{-4}	-6.5×10^{-4}
μ_{fet0} [$\text{cm}^2 \text{ V}^{-1} \text{ s}^{-1}$]	8.2×10^{-4}	8.18×10^{-4}
I_0 [A]	4.9×10^{-4}	4.9×10^{-10}
I_{0S} [A]	5.1×10^{-9}	5.1×10^{-9}
CC	6.81×10^9	6.81×10^9

4.3 Otimização dos parâmetros de ajuste da extração do UMEM no MATLAB

Após a verificação do MATLAB como uma plataforma com implementação satisfatória do modelo e método de extração do UMEM, a próxima etapa consistiu em aprimorar o código no MATLAB para dados de entrada genéricos, como descrito no capítulo 3. Uma das vantagens obtidas através dessas modificações é a visualização das alterações nas funções de ajustes e os parâmetros extraídos do modelo a cada simulação. Assim, é possível aplicar o algoritmo descrito no capítulo anterior, otimizando a extração de parâmetros do UMEM.

Para esta seção são utilizados dados experimentais de dois transistores, TUD OTFT e TUD VOTFT. O CINVESTAV OTFT da seção anterior não foi analisado, pois seus parâmetros de ajuste já haviam sido escolhidos no código em MATHCAD. Assim, inicia-se a análise pelas curvas e parâmetros obtidos para o TUD OTFT.

4.3.1 Otimização dos parâmetros de ajuste da extração para o TUD OTFT

Após três simulações, não foi observada uma melhoria do ajuste nas curvas de transferências, mostradas nos gráficos III.1, III.4, III.2, III.5, III.3 e III.6 (anexos). No entanto, notou-se uma melhoria sutil das curvas de saída nos gráficos 4.4, 4.5 e 4.6, para as quais o intervalo escolhido para a segunda simulação ajustou as curvas modeladas às experimentais de forma ótima (nota-se no gráfico 4.5 que para $V_{gs} = -6V$, a curva modelada se aproxima mais dos dados experimentais entre a região linear e de saturação, assim como nota-se também um maior distanciamento entre as correntes para $V_{gs} = -2V$ e $0V$).

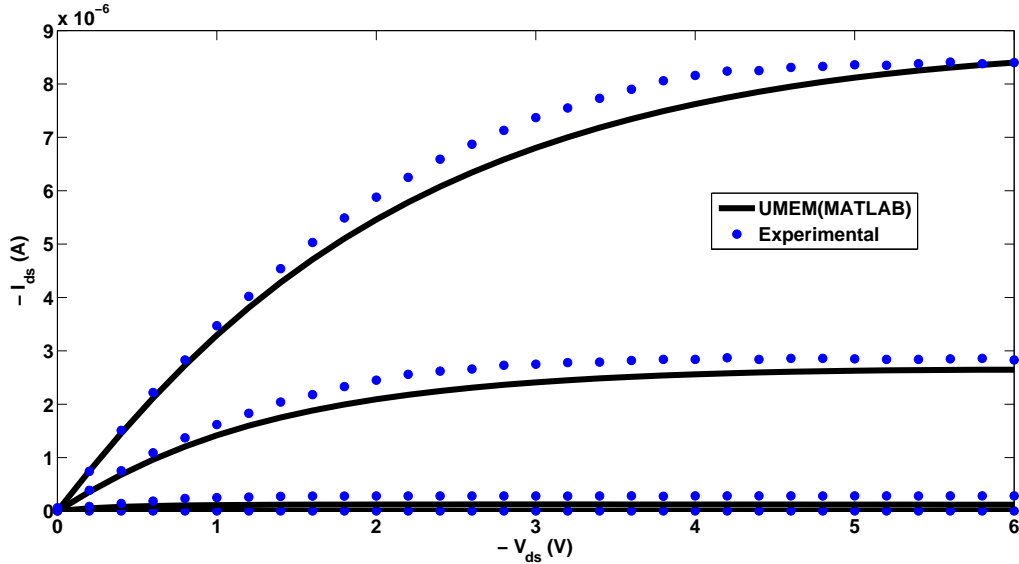


Figura 4.4: Curvas características de saída para o TUD OTFT encontradas experimentalmente e modelada pelo MATLAB usando os parâmetros da primeira simulação na tabela 4.2 para $V_{gs} = 0V, -2V, -4V$ e $-6V$. $V_{H,min} = -4V$, $V_{H,max} = -6V$, $V_{IPC,min} = -4V$, $V_{IPC,max} = -6V$.

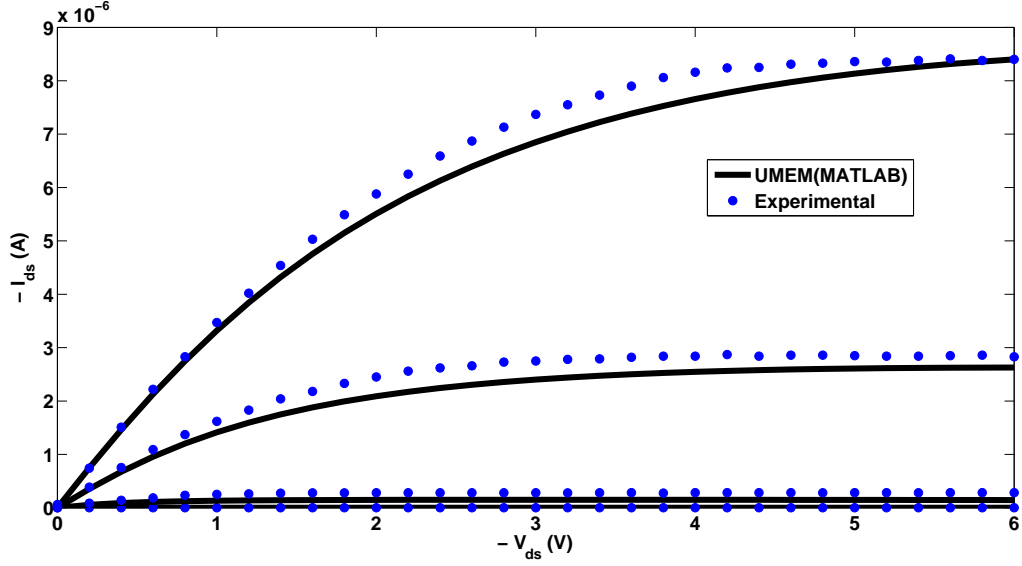


Figura 4.5: Curvas características de saída para o TUD OTFT encontradas experimentalmente e modelada pelo MATLAB usando os parâmetros da segunda simulação na tabela 4.2 com $V_{gs} = 0V, -2V, -4V$ e $-6V$. $V_{H,min} = -3V$, $V_{H,max} = -5V$, $V_{IPC,min} = -4V$, $V_{IPC,max} = -6V$.

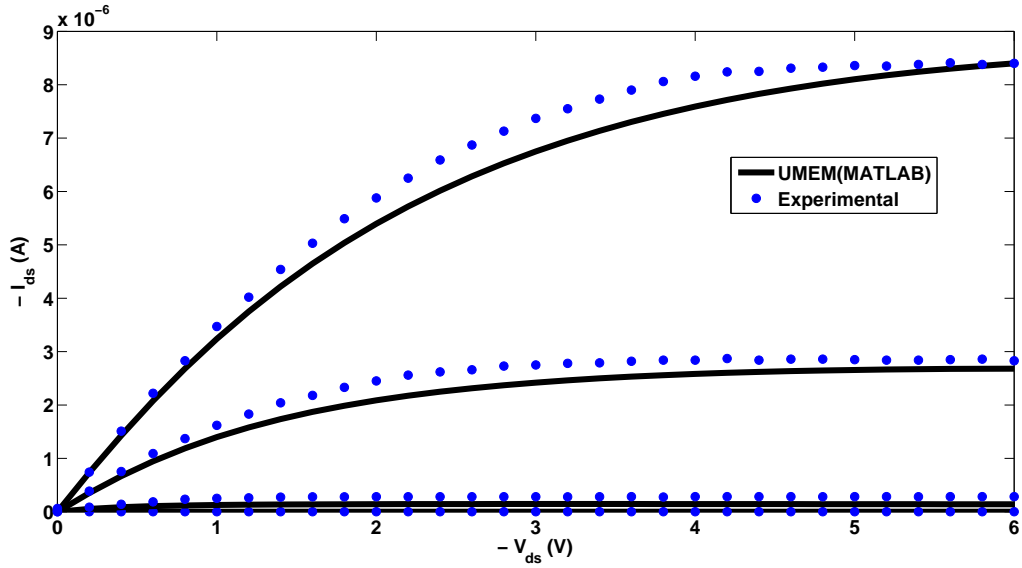


Figura 4.6: Curvas características de saída para o TUD OTFT encontradas experimentalmente e modelada pelo MATLAB usando os parâmetros da terceira simulação na tabela 4.2 com $V_{gs} = 0V, -2V, -4V$ e $-6V$. $V_{H,min} = -2V$, $V_{H,max} = -4V$, $V_{IPC,min} = -4V$, $V_{IPC,max} = -6V$.

Observa-se que a curva da função $H(V_{gs})$ para a primeira simulação, no gráfico 4.7, é uma reta inclinada (com uma região de descontinuidade) que não se altera significativamente com a mudança dos intervalos, como pode ser visto nos gráficos III.7 e III.8. Isso justifica o motivo dos parâmetros pouco serem alterados nas demais simulações, mostrados na tabela 4.2, assim como as próprias funções de ajuste $IPC(V_{gs})$ e $IICS1(V_{gs})$, que dependem dos parâmetros extraídos da função $H(V_{gs})$. Isso explica também porque não se alterou a largura dos intervalos escolhidos

para $H(V_{gs})$ nos gráficos III.7 e III.8 (anexos), assim como os passos de V_{gs} entre os intervalos ao longo das simulações. Pode ser observado que, independentemente do intervalo usado para $H(V_{gs})$, as curvas das funções de ajuste $IPC(V_{gs})$ e $IIS1(V_{gs})$ permanecem as mesmas, como mostrado nos gráficos 4.8, III.9 e III.10 (dois últimos nos anexos), já que também seus intervalos não são alterados após a primeira simulação (menor distância entre $IPC(V_{gs})$ e $IIS1(V_{gs})$ obtida na primeira simulação).

Nota-se também que o valor da resistência pode ser visto como determinante na escolha da melhor simulação, que cresce com a redução dos limites do intervalo selecionado para $H(V_{gs})$, até atingir um valor máximo e se tornar nula, como pode ser visto na tabela 4.2. Para o intervalo considerado como o melhor (segunda simulação na tabela 4.2), a resistência encontrada foi a maior possível, igual à $6.6538 \times 10^3 \Omega$.

Tabela 4.2: Parâmetros extraídos do TUD OTFT através do MATLAB para diferentes parâmetros de ajuste com intervalos de largura de 2V. Os dados da tabela correspondem às três simulações. A segunda simulação foi considerada como a melhor.

Parâmetro extraído	1ª simulação	2ª simulação	3ª simulação
V_T [V]	-1.34	-1.19	-1.28
γ_a	0.29	0.396	0.117
V_{aa} [V]	40.341	24.142	38.724
R [Ω]	1.96×10^3	6.65×10^3	0
α_s	0.7	0.66	0.72
m	1.418	1.476	1.401
λ [1/V]	-0.02	-0.02	-0.02
μ_{fet0} [$cm^2 V^{-1} s^{-1}$]	0.3425	0.2838	0.6519
Parâmetro de ajuste	1ª simulação	2ª simulação	3ª simulação
$V_{H,min}$ [V]	-4	-3	-2
$V_{H,max}$ [V]	-6	-5	-4
$V_{IPC,min}$ [V]	-4	-4	-4
$V_{IPC,max}$ [V]	-6	-6	-6

Em relação à escolha de intervalos com largura de 1V para a função $H(V_{gs})$, os parâmetros encontrados foram também semelhantes aos anteriores, como mostrado na tabela 4.3. A diferença é que ao escolher intervalos menores é possível obter um valor maior para resistência, assim como uma aproximação melhor das curvas modeladas. A maior resistência pode ser vista na tabela 4.3, igual à $8.37 \times 10^3 \Omega$. No entanto, é importante notar que apesar de intervalos menores produzirem melhores resultados, o tempo empregado com o número maior de simulações não compensa os ganhos, que não são notados facilmente nos gráficos.

Conclui-se que, considerando três simulações como quantidade satisfatória para determinar os parâmetros de ajuste da extração, escolhe-se a segunda simulação, para um intervalo de largura 2V, como a melhor para o TUD OTFT, devido a sua curva de saída que possui uma aproximação maior com os dados experimentais, para a qual os parâmetros podem ser encontrados na segunda

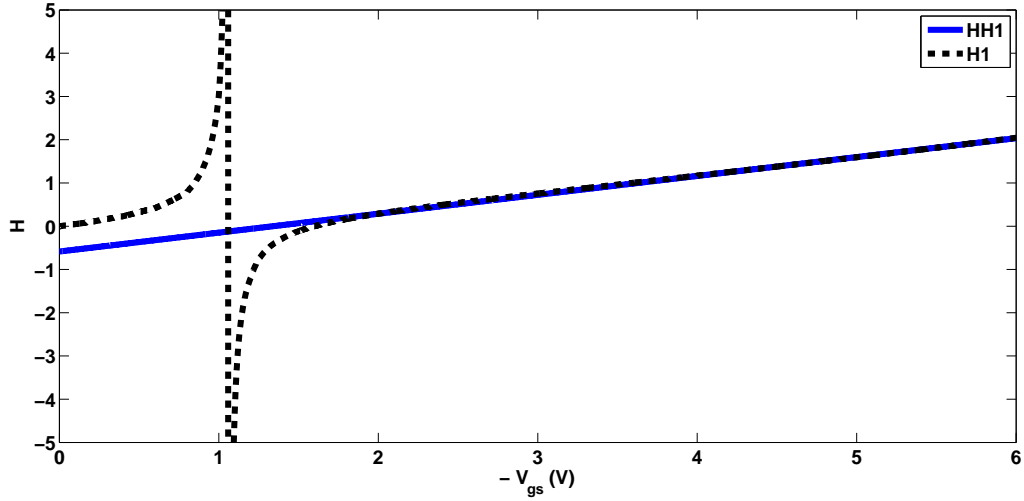


Figura 4.7: Curvas das funções de ajuste $H(V_{gs})$ e $HH1(V_{gs})$ para o TUD OTFT usando os parâmetros de ajuste da primeira simulação na tabela 4.2. $V_{H,min} = -4V$, $V_{H,max} = -6V$, $V_{IPC,min} = -4V$, $V_{IPC,max} = -6V$.

Tabela 4.3: Parâmetros extraídos do TUD OTFT através do MATLAB para diferentes parâmetros de ajuste com intervalos de largura de 1V. Os dados da tabela correspondem às quatro simulações.

Parâmetro extraído	1ª Simulação	2ª Simulação	3ª Simulação	4ª Simulação
V_T [V]	-1.34	-1.19	-1.27	-1.27
γ_a	0.215	0.362	0.416	0.172
V_{aa} [V]	74.466	27.536	22.256	438.913
R [Ω]	576.09	4.98×10^3	8.37×10^3	0
α_s	0.72	0.66	0.63	0.91
m	1.41	1.481	1.545	1.337
λ [1/V]	-0.02	-0.02	-0.02	-0.02
μ_{fet0} [$\text{cm}^2 \text{V}^{-1} \text{s}^{-1}$]	0.3965	0.3013	0.275	0.3505
Parâmetro de ajuste	1ª Simulação	2ª Simulação	3ª Simulação	4ª Simulação
$V_{H,min}$ [V]	-5	-4	-3	-2
$V_{H,max}$ [V]	-6	-5	-4	-3
$V_{IPC,min}$ [V]	-5	-5	-5	-5
$V_{IPC,max}$ [V]	-6	-6	-6	-6

coluna da tabela 4.2.

4.3.2 Otimização dos parâmetros de ajuste da extração para o TUD VOTFT

A mesma análise é feita para o TUD VOTFT. Percebe-se que para o primeiro intervalo, mostrado na primeira coluna da tabela 4.4 com os parâmetros extraídos, a função $H(V_{gs})$ é uma curva com menor linearidade, como pode ser visto no gráfico 4.9. Logo, sabe-se que os parâmetros

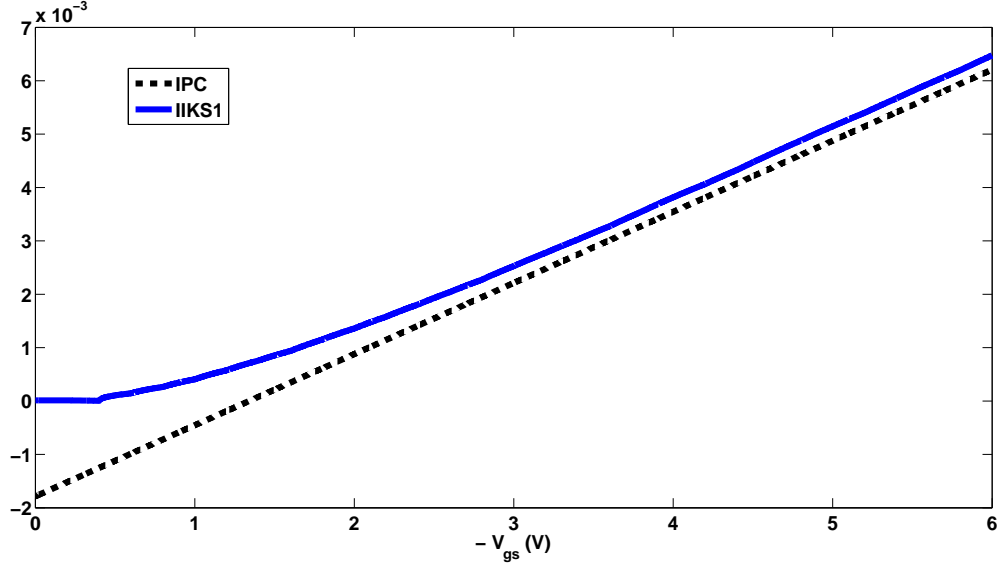


Figura 4.8: Curvas das funções de ajuste $IPC(V_{gs})$ e $IKS1(V_{gs})$ para o TUD OTFT usando os parâmetros de ajuste da primeira simulação na tabela 4.2. $V_{H,min} = -4V$, $V_{H,max} = -6V$, $V_{IPC,min} = -4V$, $V_{IPC,max} = -6V$.

devem variar mais com o intervalo escolhido se comparado ao TUD OTFT, o que pode ser notado tabela 4.4. Isso causa maiores mudanças entre as curvas de saída e transferência com cada simulação. Por esse motivo, os passos de V_{gs} entre os intervalos foi reduzido à $0.5 V$.

Através da redução dos limites do intervalo usado para a função $H(V_{gs})$, mostrado no gráfico 4.10, as curvas se aproximam mais dos dados experimentais, como pode ser observado ao comparar-se 4.15 e 4.16 (curvas de saída) ou 4.11 e 4.12 (curvas de transferências para $V_{gs} = -0.5V$) ou 4.13 e 4.14 (curvas de transferências para $V_{gs} = -5V$).

Nota-se que a terceira ou segunda simulações podem ser consideradas como resultados ótimos, pois enquanto a terceira simulação apresenta melhor ajuste aos dados experimentais na região de saturação, a segunda simulação se adequa melhor à região linear. No entanto, a terceira simulação foi escolhida como a que possui mais resultados satisfatórios, pois esta também possui uma representação mais adequada da região sublimiar na curva de transferência (como pode ser visto no gráfico 4.12).

O efeito da resistência não foi notado para o TUD VOTFT, mostrada na tabela 4.4, que atinge valor máximo após a terceira simulação.

Observa-se também que apesar do valor de V_{aa} encontrado para o TUD VOTFT na tabela 4.4 ser muito pequeno, esse parâmetro é apenas matemático, sem sentido físico quando analisado individualmente. O que realmente importa é o valor da mobilidade que é determinado através de V_{aa} e γ_a , igual à $\mu_{fet0} = 3.1048 \times 10^{-4} \text{ cm}^2 \text{ V}^{-1} \text{ s}^{-1}$.

Diferentemente do TUD OTFT, o intervalo utilizado para determinar a curva $IPC(V_{gs})$ na primeira simulação estava muito distante do ideal, como pode ser notado no gráfico 4.17. Consequentemente, as maiores diferenças entre as curvas ocorrem entre a primeira e segunda simulações,

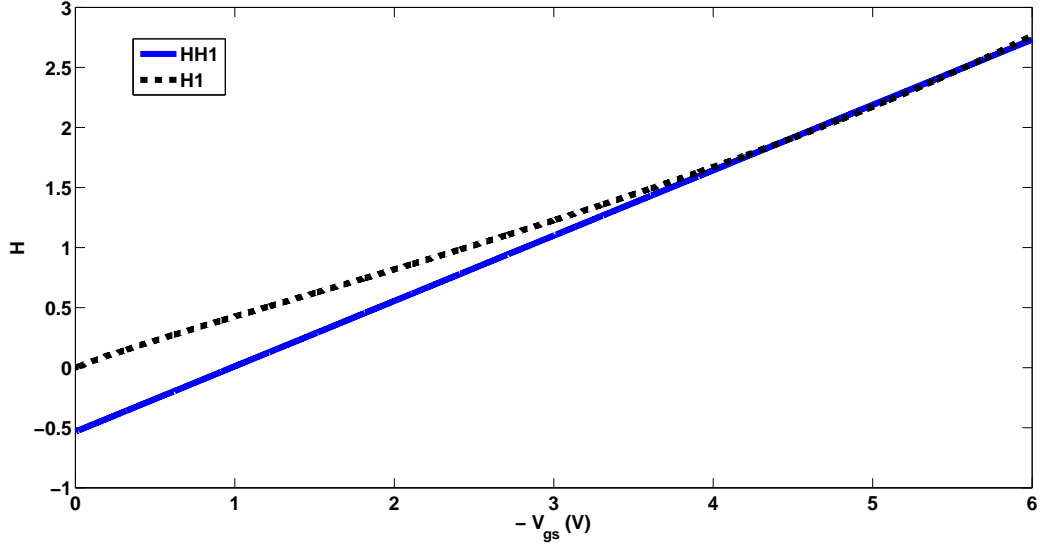


Figura 4.9: Curvas das funções de ajuste $H(V_{gs})$ e $HH1(V_{gs})$ para o TUD VOTFT usando os parâmetros de ajuste da primeira simulação na tabela 4.4. $V_{H,min} = -4V$, $V_{H,max} = -6V$, $V_{IPC,min} = -3V$, $V_{IPC,max} = -5V$.

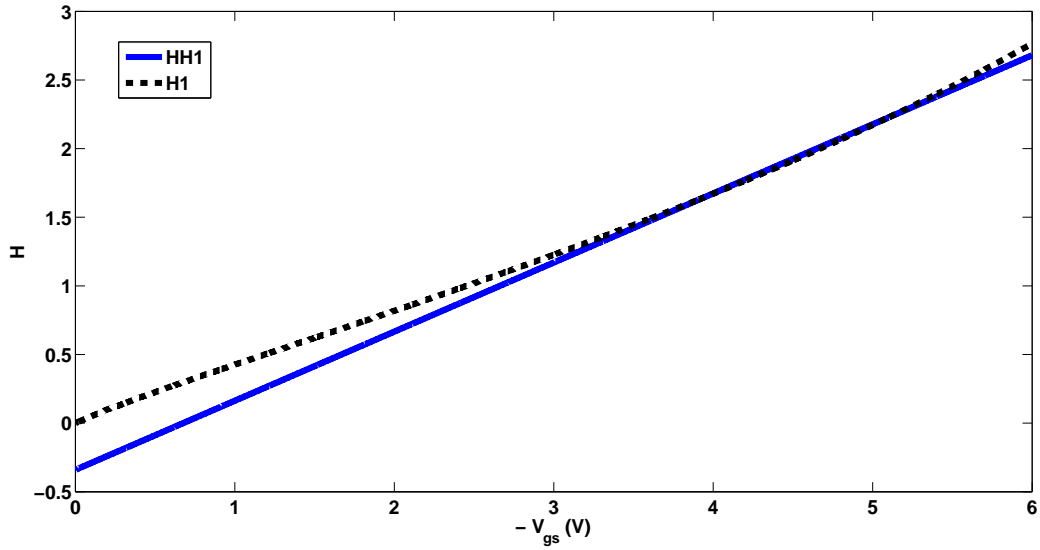


Figura 4.10: Curvas das funções de ajuste $H(V_{gs})$ e $HH1(V_{gs})$ para o TUD VOTFT usando os parâmetros de ajuste da terceira simulação na tabela 4.4. $V_{H,min} = -3.5V$, $V_{H,max} = -5.5V$, $V_{IPC,min} = -1.8V$, $V_{IPC,max} = -2.2V$.

como pode ser visto nos gráficos 4.20, 4.18 e 4.19. Após a segunda simulação, o intervalo escolhido para $IPC(V_{gs})$ não foi modificado, pois a região em que a distância entre as duas funções $IPC(V_{gs})$ e $IIKS1(V_{gs})$ é mínima já havia sido encontrada. Percebe-se, então, que as pequenas mudanças entre os próximos gráficos III.11 e III.12 ($IPC(V_{gs})$ e $IIKS1(V_{gs})$) nas segunda e terceira simulações são causadas somente pelas variações do intervalo para a função $H(V_{gs})$.

Tabela 4.4: Parâmetros extraídos do TUD VOTFT através do MATLAB para diferentes parâmetros de ajuste. Os dados da tabela correspondem às três simulações. A terceira simulação foi considerada como a melhor, para a qual os parâmetros extraídos foram usados em simulações de circuitos no QUCS (seção 4.6).

Parâmetro extraído	1ª simulação	2ª simulação	3ª simulação
V_T [V]	-0.98	-0.98	-0.68
γ_a	-0.162	-0.162	-0.014
V_{aa} [V]	1.257×10^{-21}	1.257×10^{-21}	7.431×10^{-259}
R [Ω]	1.473×10^3	1.473×10^3	3.46×10^3
α_s	1.942	1.029	1.097
m	1.122	4.746	3.269
λ [1/V]	-3.515×10^{-4}	6.127×10^{-3}	-7.775×10^{-3}
μ_{fet0} [$\text{cm}^2 \text{V}^{-1} \text{s}^{-1}$]	4.08×10^{-4}	4.08×10^{-4}	3.1048×10^{-4}
Parâmetro de ajuste	1ª simulação	2ª simulação	3ª simulação
$V_{H,\min}$ [V]	-4	-4	-3.5
$V_{H,\max}$ [V]	-6	-6	-5.5
$V_{IPC,\min}$ [V]	-3	-1.8	-1.8
$V_{IPC,\max}$ [V]	-5	-2.2	-2.2

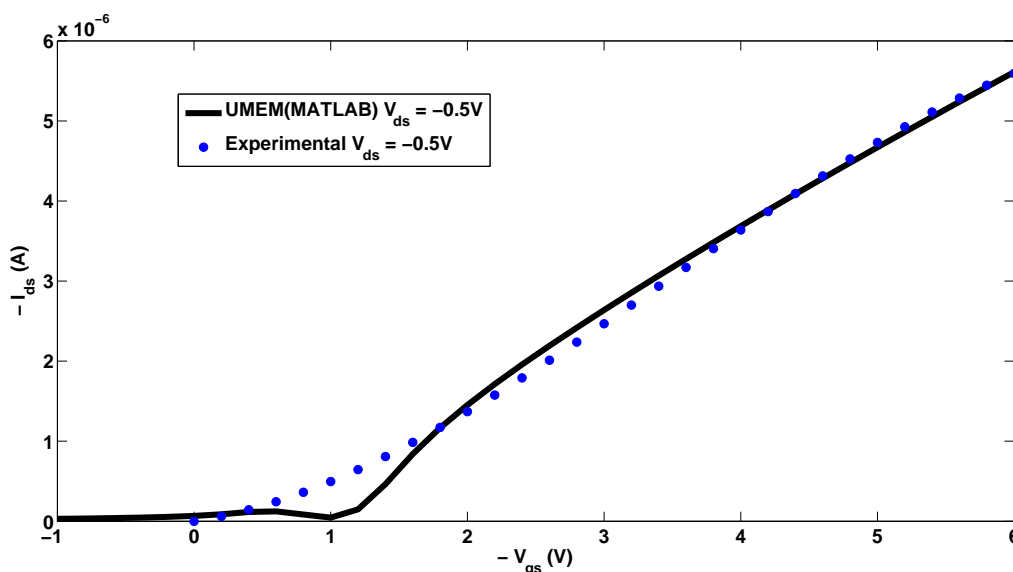


Figura 4.11: Curvas características de transferência para o TUD VOTFT encontradas experimentalmente e modelada pelo MATLAB usando os parâmetros da segunda simulação na tabela 4.4. $V_{H,\min} = -4\text{V}$, $V_{H,\max} = -6\text{V}$, $V_{IPC,\min} = -1.8\text{V}$, $V_{IPC,\max} = -2.2\text{V}$.

Cogitou-se descrever o código acima em MATLAB, sem demandar do usuário tantas simulações. No entanto, a determinação do tamanho do intervalo de V_{gs} para o cálculo de $\text{IICS1}(V_{gs})$ requer visualização das funções de ajuste. Encontrar o ponto em que a diferença entre $\text{IICS1}(V_{gs})$ e $\text{IPC}(V_{gs})$ é mínima não é suficiente. Precisa-se saber também qual será a varredura em torno

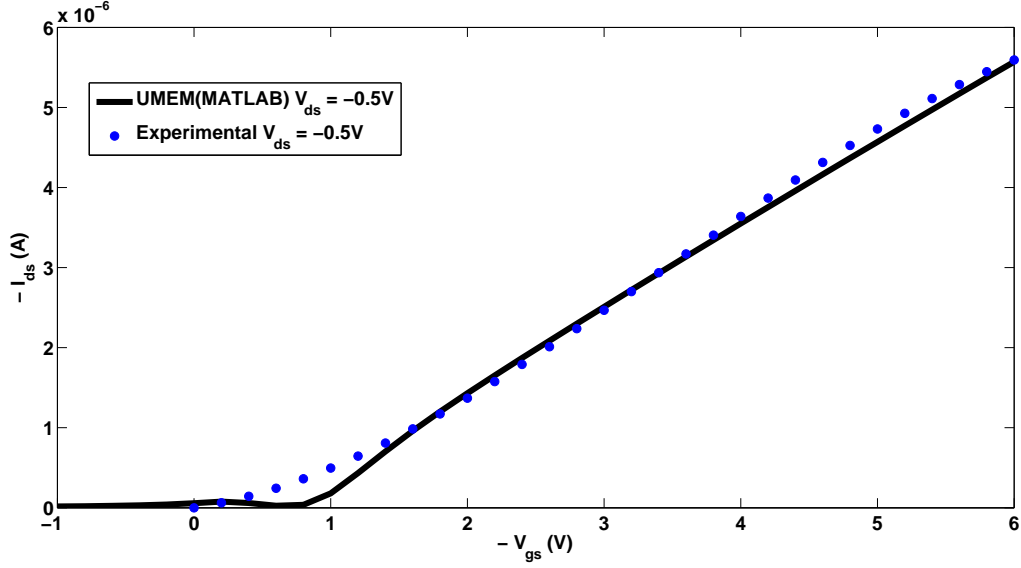


Figura 4.12: Curvas características de transferência para o TUD VOTFT encontradas experimentalmente e modelada pelo MATLAB usando os parâmetros da terceira simulação na tabela 4.4. $V_{H,min} = -3.5V$, $V_{H,max} = -5.5V$, $V_{IPC,min} = -1.8V$, $V_{IPC,max} = -2.2V$.

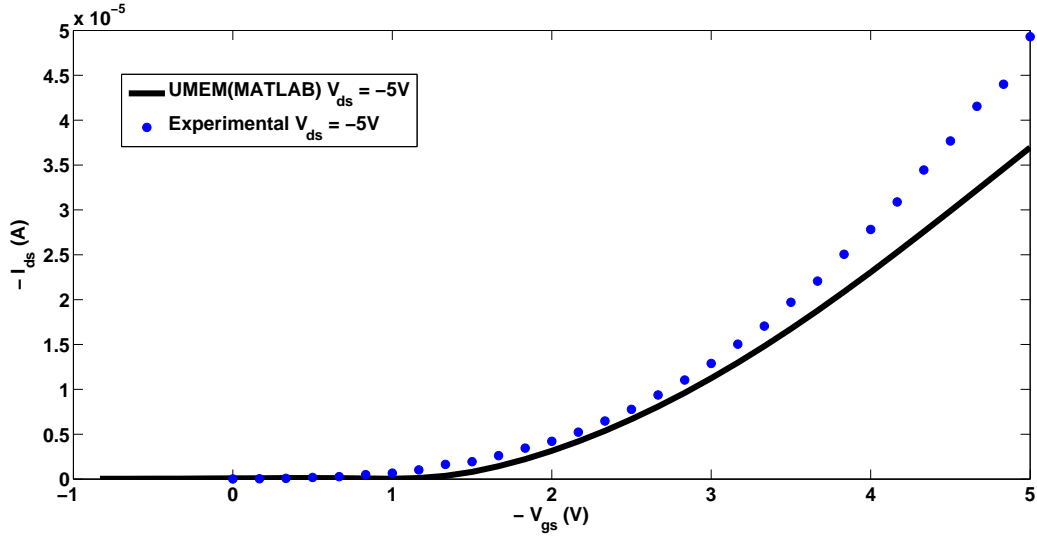


Figura 4.13: Curvas características de transferência para o TUD VOTFT encontradas experimentalmente e modelada pelo MATLAB usando os parâmetros da segunda simulação na tabela 4.4. $V_{H,min} = -4V$, $V_{H,max} = -6V$, $V_{IPC,min} = -1.8V$, $V_{IPC,max} = -2.2V$.

desse ponto, que depende dos dados experimentais. No caso acima, um intervalo entre -1.8V e -2.2V em torno do ponto de mínimo (aproximadamente -2V) mostrou ser suficiente.

Conclui-se que as curvas modeladas pelo UMEM podem ser melhoradas através de poucas simulações, ao mudar-se os intervalos de V_{gs} nos quais se calculam as funções de ajuste, seguindo os passos descritos no capítulo 3. Para o TUD VOTFT, a terceira simulação apresentou os melhores resultados, cujos parâmetros podem ser encontrados na tabela 4.4.

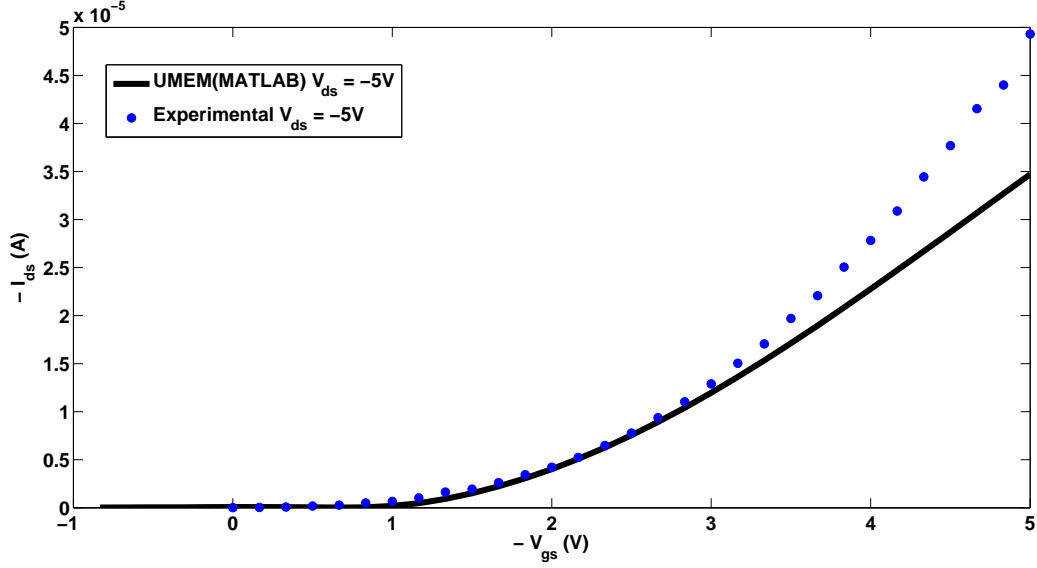


Figura 4.14: Curvas características de transferência para o TUD VOTFT encontradas experimentalmente e modelada pelo MATLAB usando os parâmetros da terceira simulação na tabela 4.4. $V_{H,min} = -3.5V$, $V_{H,max} = -5.5V$, $V_{IPC,min} = -1.8V$, $V_{IPC,max} = -2.2V$.

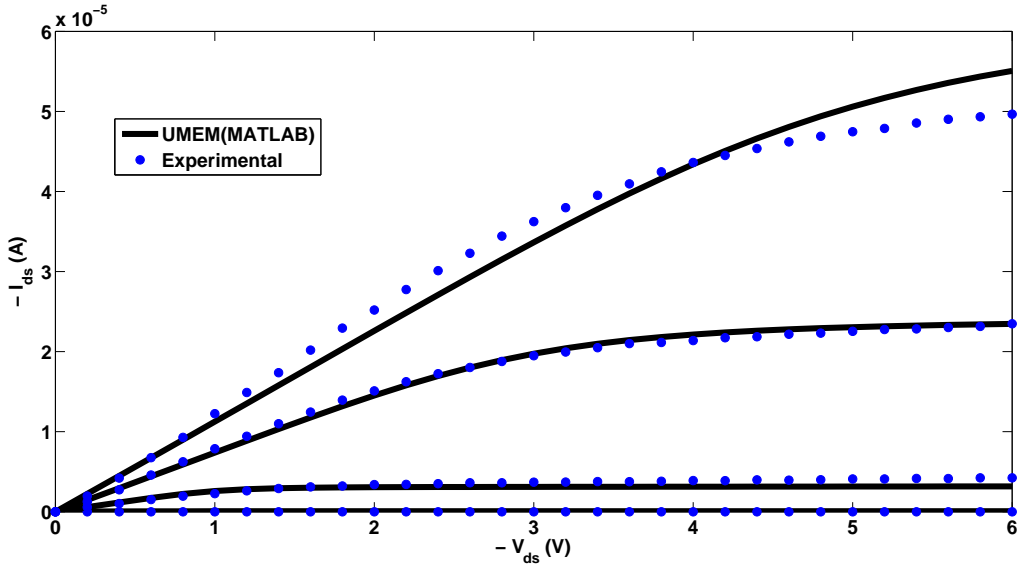


Figura 4.15: Curvas características de saída para o TUD VOTFT encontradas experimentalmente e modelada pelo MATLAB usando os parâmetros da segunda simulação na tabela 4.4 com $V_{gs} = 0V, -2V, -4V$ e $-6V$. $V_{H,min} = -4V$, $V_{H,max} = -6V$, $V_{IPC,min} = -1.8V$, $V_{IPC,max} = -2.2V$.

4.4 Otimização dos parâmetros ajustáveis do modelo UMEM

Com objetivo de obter um modelo com uma aproximação ainda melhor dos dados experimentais, os parâmetros de ajuste das curvas de saída e transferência Q , Q_S , DV , DVS , S e SS foram otimizados por dois métodos, conforme descritos no capítulo 3, a menor média e o menor valor máximo dos erros relativos das correntes. Inicia-se a análise com os resultados do TUD OTFT.

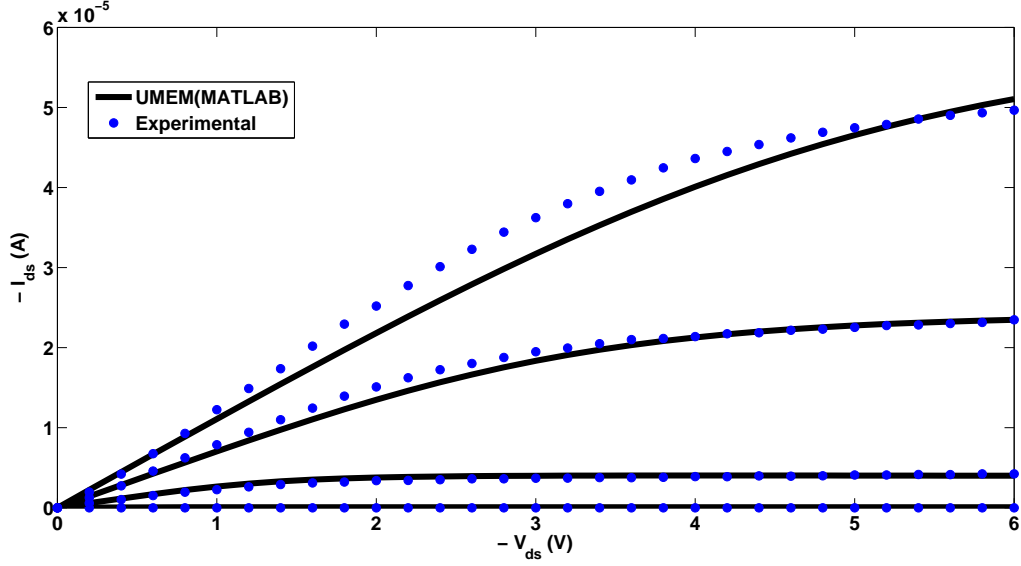


Figura 4.16: Curvas características de saída para o TUD VOTFT encontradas experimentalmente e modelada pelo MATLAB usando os parâmetros da terceira simulação na tabela 4.4 com $V_{gs} = 0V, -2V, -4V$ e $-6V$. $V_{H,min} = -3.5V$, $V_{H,max} = -5.5V$, $V_{IPC,min} = -1.8V$, $V_{IPC,max} = -2.2V$.

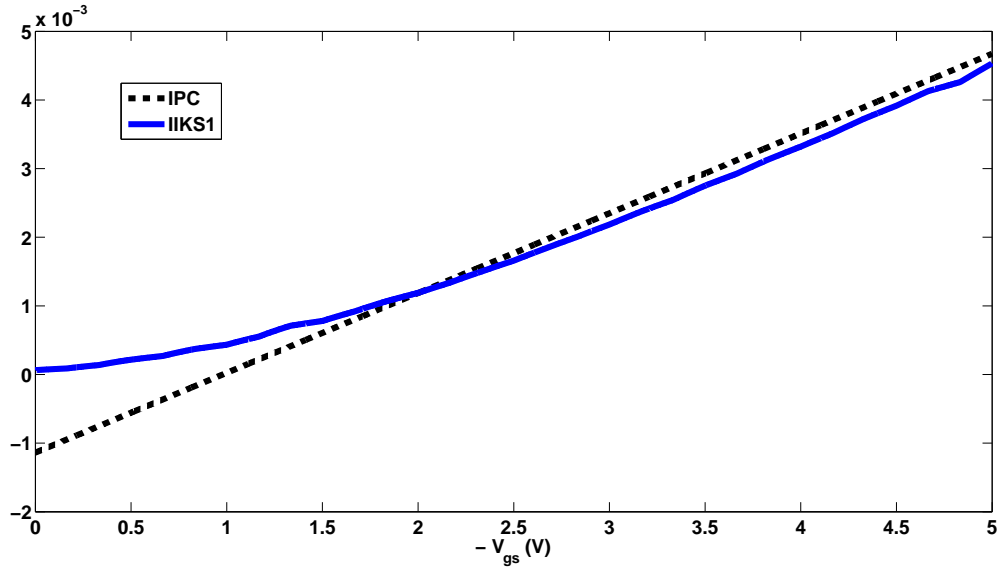


Figura 4.17: Curvas das funções de ajuste $IPC(V_{gs})$ e $IKS1(V_{gs})$ para o TUD VOTFT usando os parâmetros de ajuste da primeira simulação na tabela 4.4. $V_{H,min} = -4V$, $V_{H,max} = -6V$, $V_{IPC,min} = -3V$, $V_{IPC,max} = -5V$.

4.4.1 Otimização dos parâmetros ajustáveis do modelo para o TUD OTFT

Como pode ser visto nas gráficos 4.21a, 4.22a e 4.23a, o ajuste pelo método da menor média aos dados experimentais foi melhor em relação à escolha desses parâmetros manualmente através de várias simulações, cujos resultados são mostrados na seção anterior como segunda tentativa do TUD OTFT, nos gráficos 4.5, III.2 e III.5. No entanto, as aproximações foram iguais para

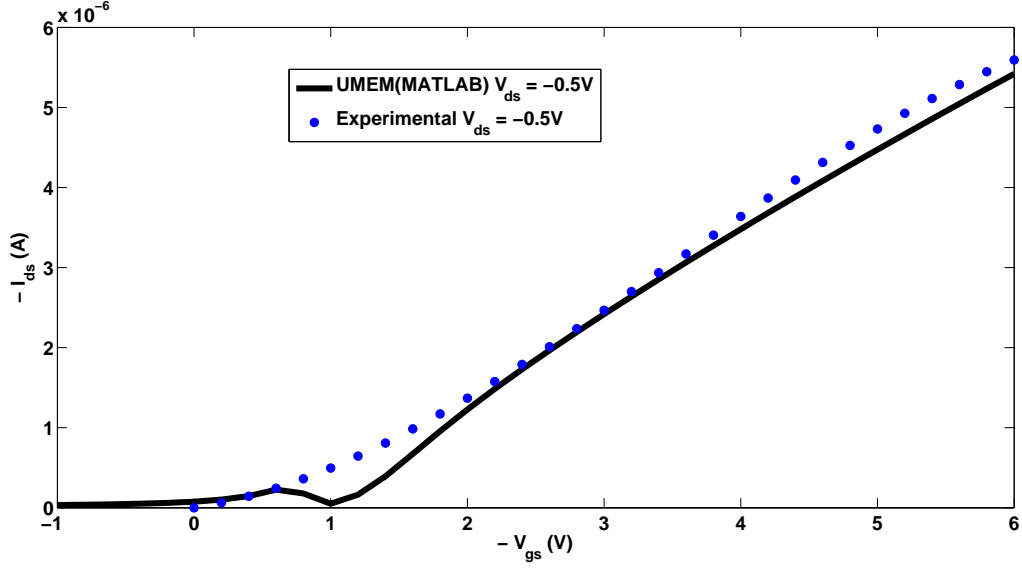


Figura 4.18: Curvas características de transferência para o TUD VOTFT encontradas experimentalmente e modelada pelo MATLAB usando os parâmetros da primeira simulação na tabela 4.4. $V_{H,min} = -4V$, $V_{H,max} = -6V$, $V_{IPC,min} = -3V$, $V_{IPC,max} = -5V$.

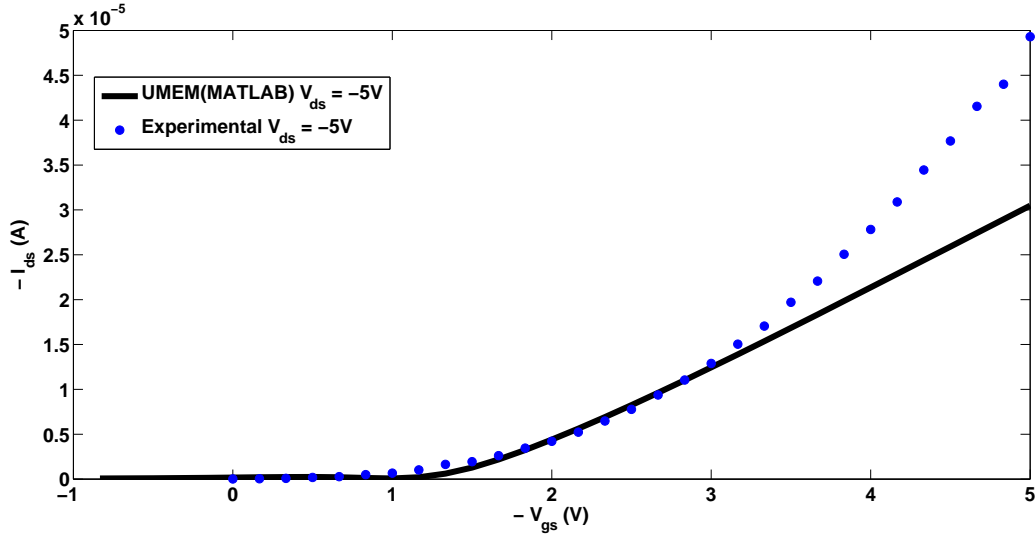


Figura 4.19: Curvas características de transferência para o TUD VOTFT encontradas experimentalmente e modelada pelo MATLAB usando os parâmetros da primeira simulação na tabela 4.4. $V_{H,min} = -4V$, $V_{H,max} = -6V$, $V_{IPC,min} = -3V$, $V_{IPC,max} = -5V$.

o método do menor valor máximo, cujos resultados podem ser vistos nos gráficos 4.21b, 4.22b e 4.23b. O que não significa que o ultimo método deva ser descartado, mas que foram obtidos valores satisfatórios para os parâmetros ajustáveis encontrados manualmente.

Comparando-se os dois métodos, percebe-se que o método do menor valor médio traz uma aproximação melhor ao longo de todos os pontos da curva, tornando-a mais próxima dos dados experimentais, enquanto o menor valor máximo suaviza as inclinações acentuadas dos gráficos,

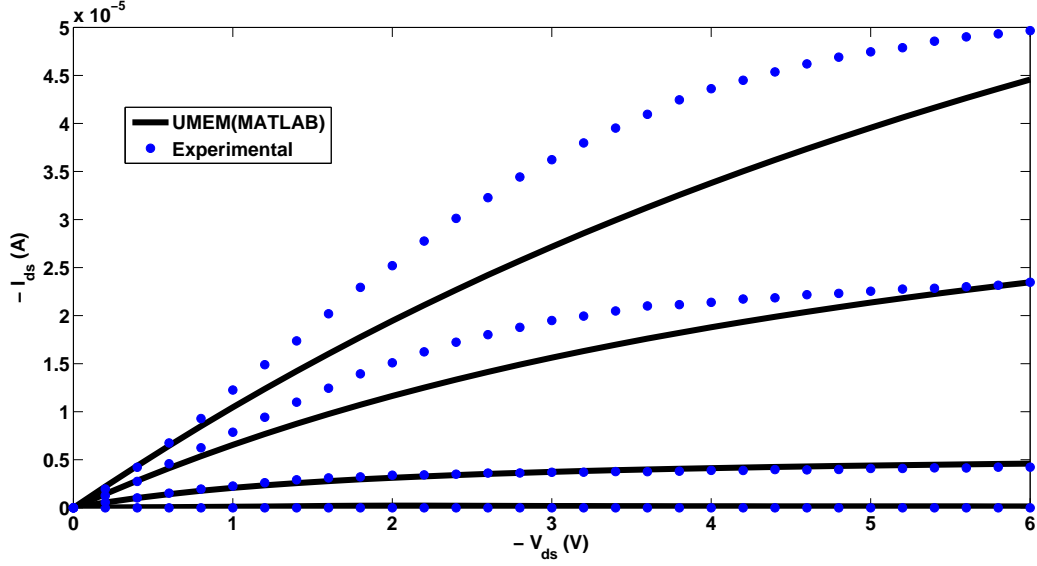
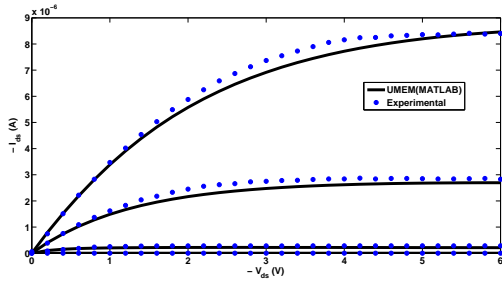
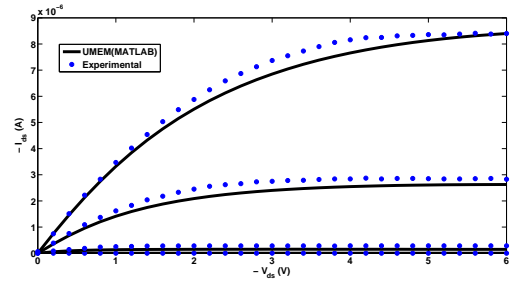


Figura 4.20: Curvas características de saída para o TUD VOTFT encontradas experimentalmente e modelada pelo MATLAB usando os parâmetros da primeira simulação na tabela 4.4 com $V_{gs} = 0V, -2V, -4V$ e $-6V$. $V_{H,min} = -4V$, $V_{H,max} = -6V$, $V_{IPC,min} = -3V$, $V_{IPC,max} = -5V$.



(a) Método da menor média

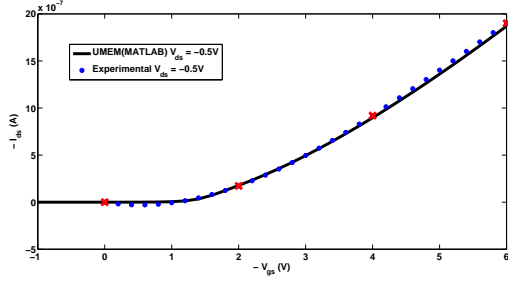


(b) Método do menor valor máximo

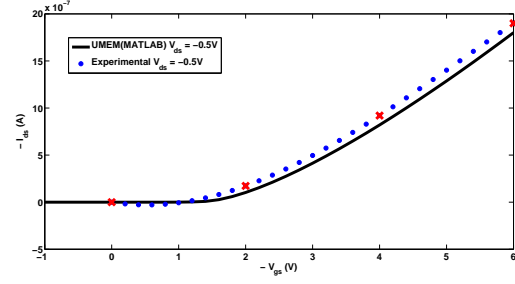
Figura 4.21: Curvas características de saída para o TUD OTFT encontradas experimentalmente e modelada pelo MATLAB usando os parâmetros ajustáveis do modelo na tabela 4.5 com $V_{gs} = 0V, -2V, -4V$ e $-6V$.

reduzindo os *spikes* na curva de transcondutâncias, como pode ser visto ao confrontar-se os gráficos 4.25a e 4.25b (curva de transcondutância pra $V_{ds} = -0.5V$), onde nota-se uma suavização do joelho entre a região de saturação e linear dos gráficos. No entanto, nem sempre o gráfico de transcondutância encontrado através do método do menor valor máximo pode ser considerado o melhor. O motivo é que a transição entre duas regiões de operação do transistor pode ocorrer de maneira bem suave, de forma que uma aproximação dos pontos experimentais ao longo de toda a curva produz uma resposta igualmente satisfatória para a transcondutância. É o que ocorre entre os gráficos 4.24a e 4.24b (curva de transcondutância pra $V_{ds} = -6V$), onde a transição da região sublimiar para a saturação ocorre de maneira bem suave e a curva de transcondutância pode ser melhor representada através do método da menor média.

Observa-se nas curvas dos gráficos 4.25a e 4.25b que a aproximação da transcondutância para

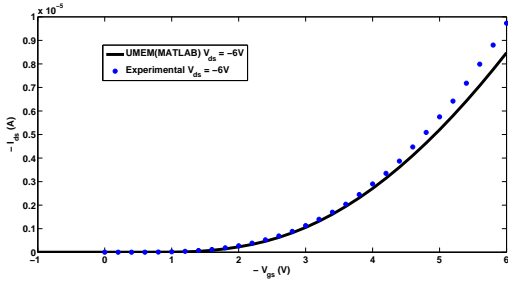


(a) Método da menor média

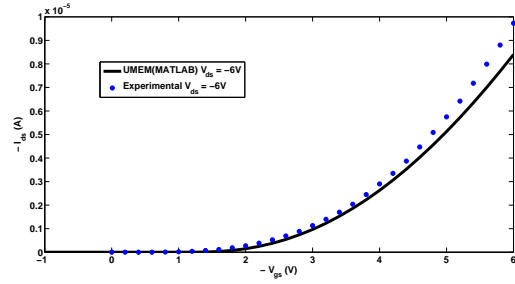


(b) Método do menor valor máximo

Figura 4.22: Curvas características de transferência para o TUD OTFT encontradas experimentalmente e modeladas pelo MATLAB usando os parâmetros ajustáveis do modelo na tabela 4.5. Nos gráficos, $V_{gs,SAT} = -1.95V$, $V_T = -1.1932V$. Os pontos correspondentes às medições estão em vermelho. Os demais, em azul, são resultados da interpolação.

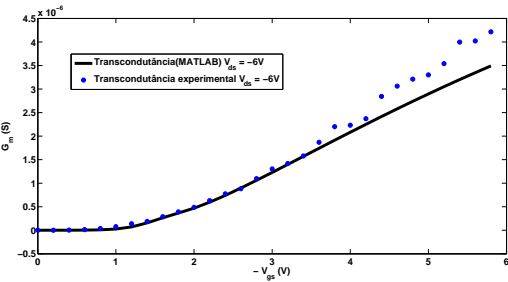


(a) Método da menor média

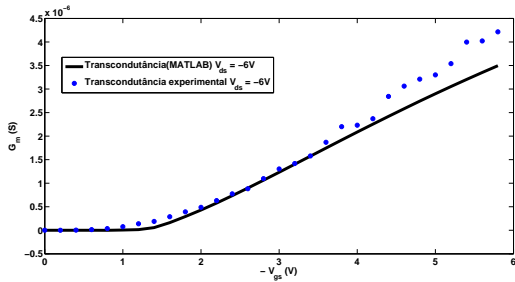


(b) Método do menor valor máximo

Figura 4.23: Curvas características de transferência para o TUD OTFT encontradas experimentalmente e modeladas pelo MATLAB usando os parâmetros ajustáveis do modelo na tabela 4.5. Nos gráficos, $V_{gs,SAT} = -10.31V$, $V_T = -1.1932V$.



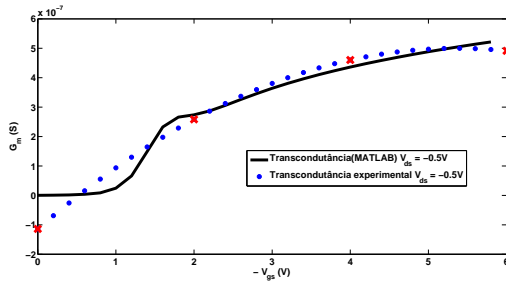
(a) Método da menor média



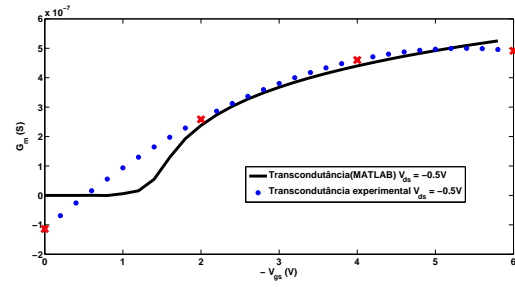
(b) Método do menor valor máximo

Figura 4.24: Curvas de transcondutância para o TUD OTFT encontradas experimentalmente e modeladas pelo MATLAB usando os parâmetros ajustáveis do modelo na tabela 4.5. Nos gráficos, $V_{gs,SAT} = -10.31V$, $V_T = -1.1932V$.

$|V_{gs}| < 2V$ não se assemelha aos dados experimentais. Isso se justifica pelo fato da curva experimental de transferência para $V_{ds} = -0.5V$ ter sido obtida através de uma interpolação cúbica a partir de 4 pontos (medidas feitas para $V_{gs} = 0V, -2V, -4V$ e $-6V$). Isso resulta em uma região entre $-2V$ e $0V$ aproximada de maneira incorreta por uma curva cúbica, atingindo valores



(a) Método da menor média



(b) Método do menor valor máximo

Figura 4.25: Curvas de transcondutâncias para o TUD OTFT encontradas experimentalmente e modeladas pelo MATLAB usando os parâmetros ajustáveis do modelo na tabela 4.5. Nos gráficos, $V_{gs,SAT} = -1.95V$, $V_T = -1.1932V$. A curva de transcondutância experimental é obtida a partir da derivada da curva interpolada de transferência. Os pontos em vermelho indicam as tensões nas quais a corrente foi medida.

negativos, como pode ser visto nos gráficos 4.22a ou 4.22b, quando na verdade a corrente nessa região decresce assintoticamente. Os gráficos modelados para transcondutância, então, podem ser considerados mais próximos do esperado do que aqueles correspondentes aos dados interpolados. No entanto, o mesmo não ocorre para as curvas de transcondutância correspondentes à $V_{ds} = -6V$, pois uma quantidade muito maior de medidas foi tomada ao longo de toda a curva, incluindo a região sublimiar.

Uma comparação entre os valores dos parâmetros ajustáveis do modelo encontrados manualmente, através dos métodos da menor média e do menor valor máximo são compilados na tabela 4.5.

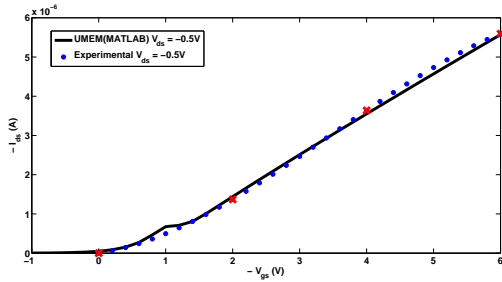
Tabela 4.5: Parâmetros ajustáveis do modelo para o TUD OTFT encontrados manualmente e otimizados pelo MATLAB através dos métodos da menor média e menor valor máximo. Os parâmetros encontrados manualmente são utilizados para validação do modelo no QUCS (seção 4.5). Os parâmetros encontrados pelo método da menor média foram considerados melhores.

Parâmetro de ajuste	Manualmente	Menor média	Menor valor máximo
Q	3	2.9	9.9
QS	3	2.9	9.9
DV	0.1	0.3	0.1
DVS	0.1	0.3	0.1
S	3	0.4	0.2
SS	3	0.4	0.2

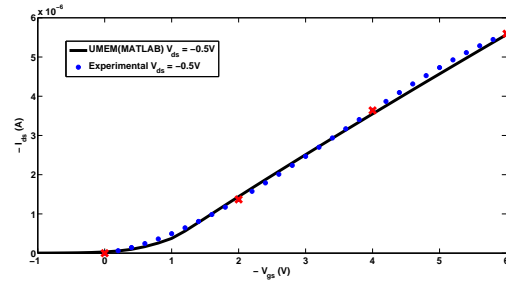
4.4.2 Otimização dos parâmetros ajustáveis do modelo para o TUD VOTFT

Uma análise semelhante pode ser feita para os resultados do TUD VOTFT. Nota-se que a otimização pelos dois métodos não alterou as curvas de saída nos gráficos IV.1 e IV.2 em relação

à determinação dos parâmetros manualmente (gráfico 4.16). As curvas de transferência para $V_{ds} = -5V$, nos gráficos 4.27b e 4.27a, não mudaram também em comparação ao gráfico 4.14, exceto por uma região próxima ao sublimiar, cujos poucos pontos foram levemente ajustados à curva experimental. No entanto, as curvas de transferência para $V_{ds} = -0.5V$ nos gráficos 4.26b e 4.26a apresentaram diferenças consideráveis comparadas ao gráfico 4.12. A primeira pode ser notada para $|V_{gs}| < 2V$, onde a curva otimizada pelo método do menor valor máximo se aproxima da curva experimental suavemente, enquanto a curva gerada pelo método da menor média cria um pico nessa região. Similarmente ao TUD OTFT, o método do menor valor máximo apresenta melhor transcondutância devido a suavização das transições bruscas entre duas regiões de operação, como pode ser percebido ao comparar os gráficos 4.28a e 4.28b (transcondutâncias para TUD VOTFT com $V_{ds} = -0.5V$).

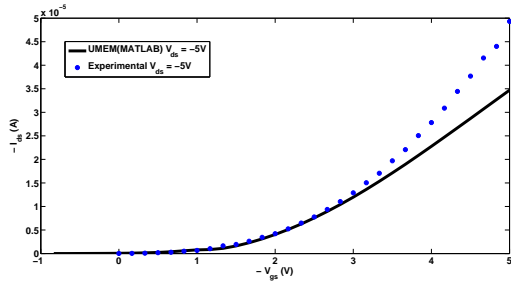


(a) Método da menor média

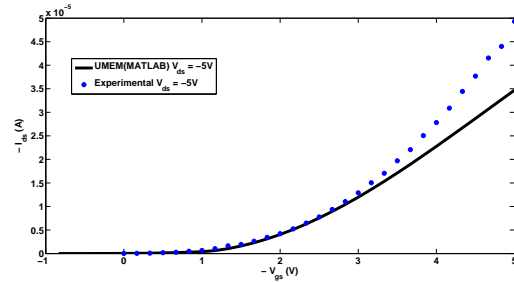


(b) Método do menor valor máximo

Figura 4.26: Curvas características de transferência para o TUD VOTFT encontradas experimentalmente e modeladas pelo MATLAB usando os parâmetros ajustáveis do modelo na tabela 4.6. Nos gráficos, $V_{gs,SAT} = -1.13V$, $V_T = -0.676V$. Os pontos correspondentes às medições estão em vermelho. Os demais, em azul, são resultados da interpolação.



(a) Método da menor média



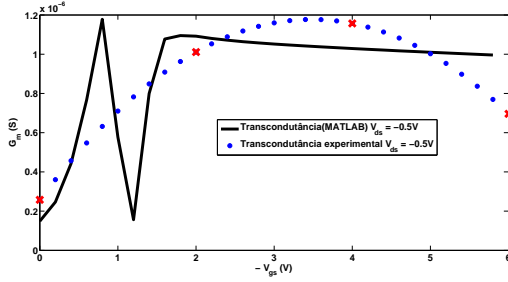
(b) Método do menor valor máximo

Figura 4.27: Curvas características de transferência para o TUD VOTFT encontradas experimentalmente e modeladas pelo MATLAB usando os parâmetros ajustáveis do modelo na tabela 4.6. Nos gráficos, $V_{gs,SAT} = -5.23V$, $V_T = -0.676V$.

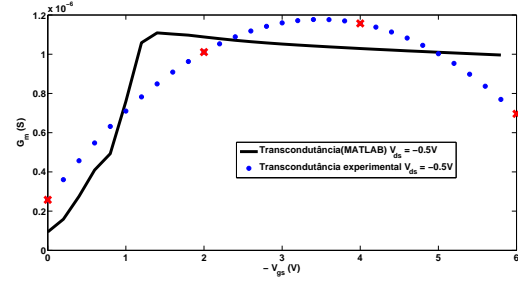
Nos dois métodos, percebe-se que há um deslocamento da curva na região de sublimiar em direção aos dados experimentais. Assim como no TUD OTFT, a interpolação cúbica foi feita a partir de poucos pontos (dados coletados em $V_{gs} = 0V$, $-2V$, $-4V$ e $-6V$) para a curva de transferência com $V_{ds} = -0.5V$, resultando em uma aproximação indevidamente linear entre $V_{gs} = 0V$ e $V_{gs} = -2V$ para a corrente experimental, que deveria decrescer assintoticamente. Em comparação

com o TUD OTFT, os erros introduzidos no TUD VOTFT são ainda maiores, por causa do alto valor da corrente em $V_{gs} = -2V$, que ao ser interpolada entre $V_{gs} = 0V$ e $V_{gs} = -2V$, apresenta uma diferença maior entre o esperado (comportamento assintótico) e o aproximado (comportamento linear). O resultado disso é que o ajuste da curva de transferência após otimização se torna pior do que o obtido com a escolha manual dos parâmetros.

Observando-se os gráficos de transcondutância para $V_{ds} = -0.5V$, 4.28a e 4.28b, nota-se que as curvas modeladas e experimentais são completamente diferentes. A causa disso é o pequeno número de pontos coletados correspondes a dados reais do experimento em torno da região sublimiar, que torna a aproximação da curva experimental de transcondutância muito distante da esperada para $|V_{gs}| < 2V$. Também pode ser notado a forma parabólica dessa curva, que é causada pela aproximação dos poucos pontos que foram digitalizados da literatura em [12] por uma função cúbica. Como a digitalização é muito vulnerável a erros de flutuação, a interpolação resulta em uma aproximação aceitável para a magnitude da corrente (os erros de flutuação são pequenos), mas uma derivada com forma parabólica que se afasta da transcondutância esperada (a curva deveria se manter constante após a transição da região de saturação para a linear). Esse comportamento parabólico não é tão evidente nas curvas de transcondutância, nos gráficos 4.25a e 4.25b, para o TUD OTFT, pois os dados desse transistor foram obtidos diretamente de um experimento, que apresentou flutuações muito menores (os três pontos da curva experimental, em $V_{gs} = -2V$, $-4V$ e $-6V$, nos gráficos 4.22a e 4.22b se aproximam mais de uma reta).



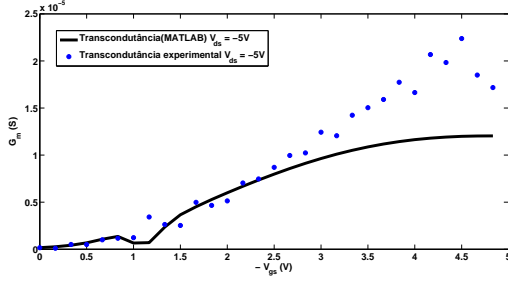
(a) Método da menor média



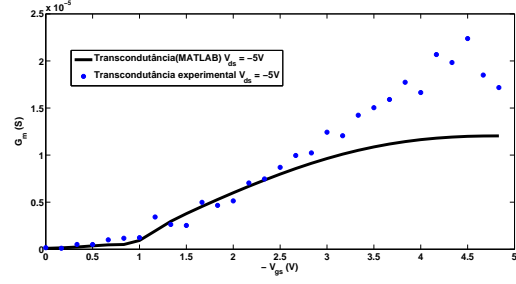
(b) Método do menor valor máximo

Figura 4.28: Curvas de transcondutância para o TUD VOTFT encontradas experimentalmente e modeladas pelo MATLAB usando os parâmetros ajustáveis do modelo na tabela 4.6. Nos gráficos, $V_{gs,SAT} = -1.13V$, $V_T = -0.676V$. A curva de transcondutância experimental é obtida a partir da derivada da curva interpolada de transferência. Os pontos em vermelho indicam as tensões nas quais a corrente foi medida.

Erros de flutuação podem ser observados também nas curvas experimentais de transcondutância do TUD VOTFT para $V_{ds} = -5V$ nos gráficos 4.29a e 4.29b, através dos pontos espaçados ao longo de toda a curva, assim como na suavidade da transição entre as regiões sublimiar e de saturação das curvas modeladas. Para essas curvas, no entanto, como muitos pontos foram digitalizados, o resultado da interpolação é satisfatório, como pode ser percebido ao se comparar as transcondutâncias modeladas e experimentais desses gráficos, que somente se distanciam a partir do ponto no qual as curvas de transferência correspondentes, nos gráficos 4.27a e 4.27b, respectivamente, também se afastam.



(a) Método da menor média



(b) Método do menor valor máximo

Figura 4.29: Curvas de transcondutâncias para o TUD VOTFT encontradas experimentalmente e modeladas pelo MATLAB usando os parâmetros ajustáveis do modelo na tabela 4.6. Nos gráficos, $V_{gs,SAT} = -5.23V$, $V_T = -0.676V$.

É importante notar que a interpolação inadequada da curva experimental de transferência para $V_{ds} = -0.5V$ também afeta a extração dos parâmetros, já que o cálculo das integrais para a corrente na região $|V_{gs}| < 2V$ assume valores maiores, alterando a função $H(V_{gs})$, assim como os parâmetros extraídos.

Uma comparação entre os valores dos parâmetros ajustáveis do modelo encontrados manualmente, através dos métodos da menor média e do menor valor máximo são compilados na tabela 4.6.

Tabela 4.6: Parâmetros ajustáveis do modelo para o TUD VOTFT encontrados manualmente e otimizados pelo MATLAB através dos métodos de menor média e menor valor máximo. Os parâmetros encontrados manualmente são utilizados para validação do modelo no QUCS (seção 4.5). Os parâmetros encontrados pelo método do menor valor máximo são utilizados nas simulações de circuitos (seção 4.6).

Parâmetro de ajuste	Manualmente	Menor média	Menor valor máximo
Q	2	5.6	4.3
QS	2	5.6	4.3
DV	0.1	0.4	0.3
DVS	0.1	0.4	0.3
S	5	0.8	0.8
SS	5	0.8	0.8

Conclui-se dessa seção que quanto menores os erros de flutuação e interpolação, maiores são as vantagens em usar os parâmetros encontrados através do método da menor média. Isso ocorre porque a curva modelada apresenta transições abruptas entre diferentes regiões de operação ao se ajustar aos dados experimentais que, devido à interpolação inadequada ou flutuações excessivas, não representam corretamente os dados reais. Utilizando esse critério, escolhe-se o método da menor média como o mais satisfatório para o TUD OTFT, cujos parâmetros ajustáveis Q , DV , S podem ser obtidos na segunda coluna da tabela 4.5. Para o TUD VOTFT, o método do menor valor máximo foi considerado o mais apropriado, para o qual os parâmetros ajustáveis do modelo

podem ser encontrados na terceira coluna da tabela 4.6.

4.5 Validação da implementação do modelo UMEM na plataforma QUCS

Com o objetivo de utilizar os dispositivos modelados pelo UMEM em um simulador de circuitos, verifica-se também se as alterações do código em Verilog A mencionadas no capítulo 3 representam corretamente as curvas características de saída e transferência do UMEM. Para isso, compara-se as curvas simuladas através do MATLAB com as produzidas pelo código em Verilog A no QUCS, avaliando se as considerações feitas para aproximar as correntes de saída por uma única equação, independentemente das condições de polarização, são aceitáveis. Para esta seção são utilizados os dados experimentais dos três transistores mencionados anteriormente: TUD OTFT, CINVESTAV OTFT e TUD VOTFT. Nessa seção, são utilizados como parâmetros de ajuste da extração aqueles considerados melhores anteriormente, ou seja, os parâmetros resultantes da segunda simulação na tabela 4.2 para o transistor TUD OTFT e os parâmetros da terceira simulação da tabela 4.4 para o transistor TUD VOTFT. Os parâmetros ajustáveis do modelo utilizados são aqueles determinados manualmente nas tabelas 4.5 e 4.6, respectivamente. Para o transistor CINVESTAV OTFT, todos os parâmetros (ajustáveis do modelo e de ajuste da extração) usados foram aqueles escolhidos no código em MATHCAD.

As curvas características de saída para os três transistores se ajustam bem entre as duas plataformas (MATLAB e QUCS), como pode ser visto nos gráficos 4.30a, 4.30b e 4.30c. Isso é esperado, pois a equação da corrente para as curvas características de saída em Verilog A é mantida igual a recomendada pela literatura quando se considera que I_0 e I_{0S} sejam nulas.

No entanto, sabe-se que ao considerar-se as correntes de fuga iguais a zero, as curvas características de transferência são afetadas, o que pode ser observado em 4.32 e 4.31 (curvas de transferência do CINVESTAV OTFT), que apresentam um leve deslocamento entre as curvas modeladas nas duas plataformas. Pode-se perceber mais nitidamente essa diferença em 4.33 (curva de transferência do TUD OTFT para $V_{ds} = -0.5V$). As demais curvas de transferência, 4.34a (curva de transferência do TUD OTFT para $V_{ds} = -6V$), 4.34c e 4.34b (curvas de transferência do TUD VOTFT) não alteraram. Essas diferenças podem resultar tanto do termo adicionado às curvas de transferência $\frac{V_{ds}}{CC}$ ou do deslocamento ao considerar-se I_0 e I_{0S} iguais a zero. No entanto, como o fator de correção $\frac{V_{ds}}{CC}$ é apenas significativo para baixos valores de corrente, o qual somente pode ser notado em regiões próximas da origem em uma escala logarítmica, pode-se afirmar que essas diferenças resultam principalmente das correntes de fuga.

Nota-se que a aproximação feita pelo código em Verilog A, ao desconsiderar I_0 , torna a corrente simulada pelo QUCS em 4.33 (curva de transferência do TUD OTFT para $V_{ds} = -0.5V$) mais próxima dos dados experimentais quando comparada com a modelada pelo MATLAB. Isso se justifica pela forma como a corrente de fuga é encontrada para $V_{ds} = -0.5V$. É feita uma extrapolação dos dados experimentais para valores positivos de V_{gs} , já que nessa região não são realizadas medições. O resultado disso é uma reflexão da curva experimental em torno do eixo das ordenadas e o valor

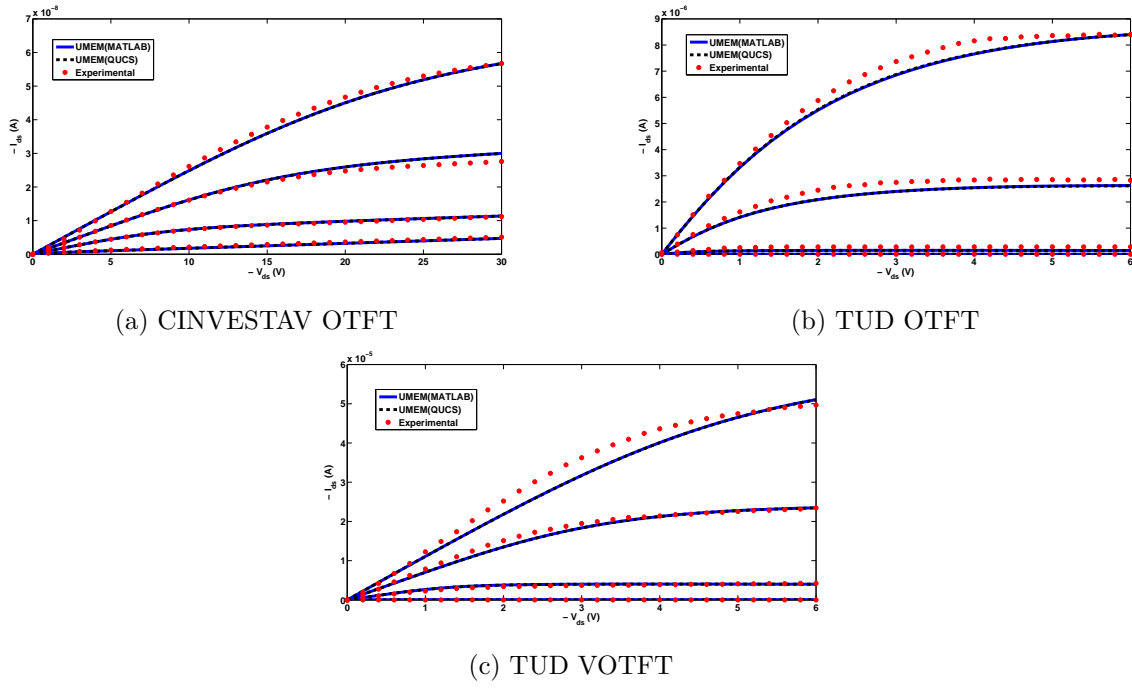


Figura 4.30: Curvas características de saída encontradas experimentalmente e modeladas pelo MATLAB e QUCS para $V_{gs} = 0V, -10V, -20V$ e $-30V$.

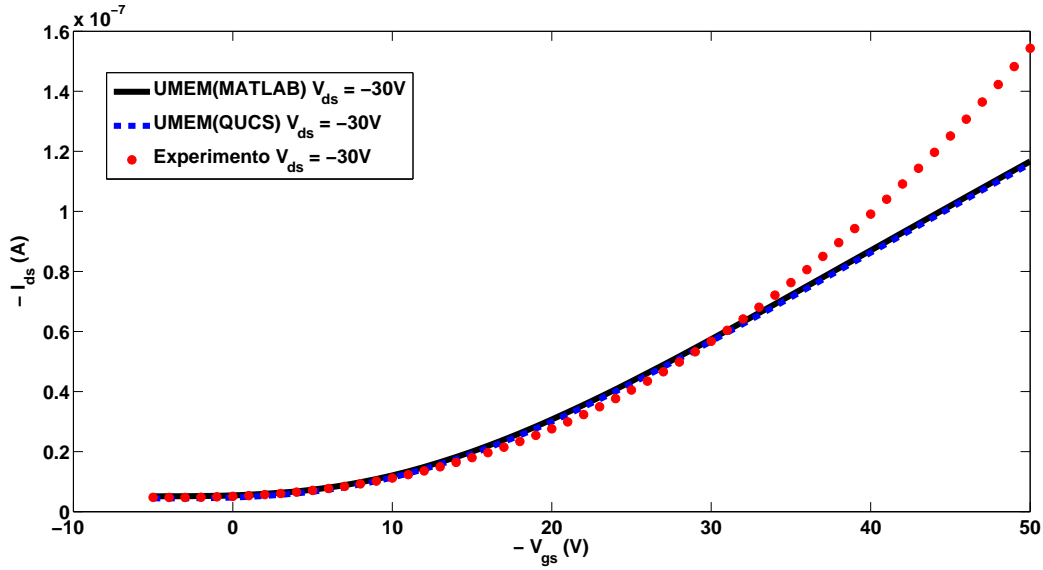


Figura 4.31: Curvas características de transferência para o CINVESTAV OTFT encontradas experimentalmente e modeladas pelo MATLAB e QUCS para $V_{ds} = -30V$.

estimado para I_0 distancia-se muito do real, que é próximo de zero. No entanto, não se nota o mesmo efeito na curva equivalente para o CINVESTAV OTFT, no gráfico 4.32, que possui V_T positivo, portanto corrente diferente de zero para $V_{gs} = 0V$. Logo, o erro resultante ao estimar-se I_0 se torna menor e a curva modelada pelo MATLAB, com $I_0 \neq 0$, mais próxima da experimental.

No entanto, mesmo que o código em Verilog A possua uma aproximação pior nesse ultimo caso

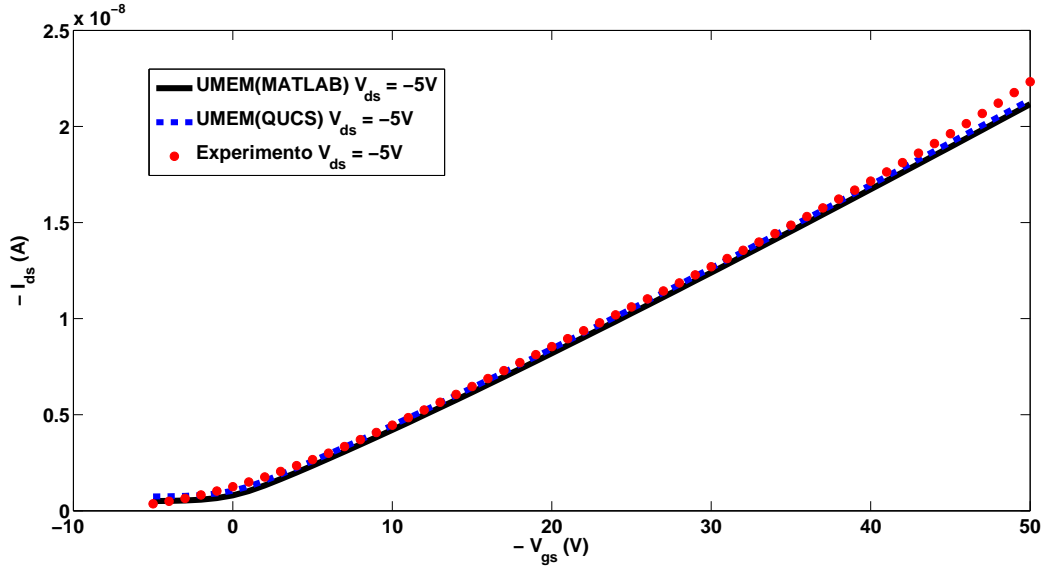


Figura 4.32: Curvas características de transferência para o CINVESTAV OTFT encontradas experimentalmente e modeladas pelo MATLAB e QUCS para $V_{ds} = -5V$.

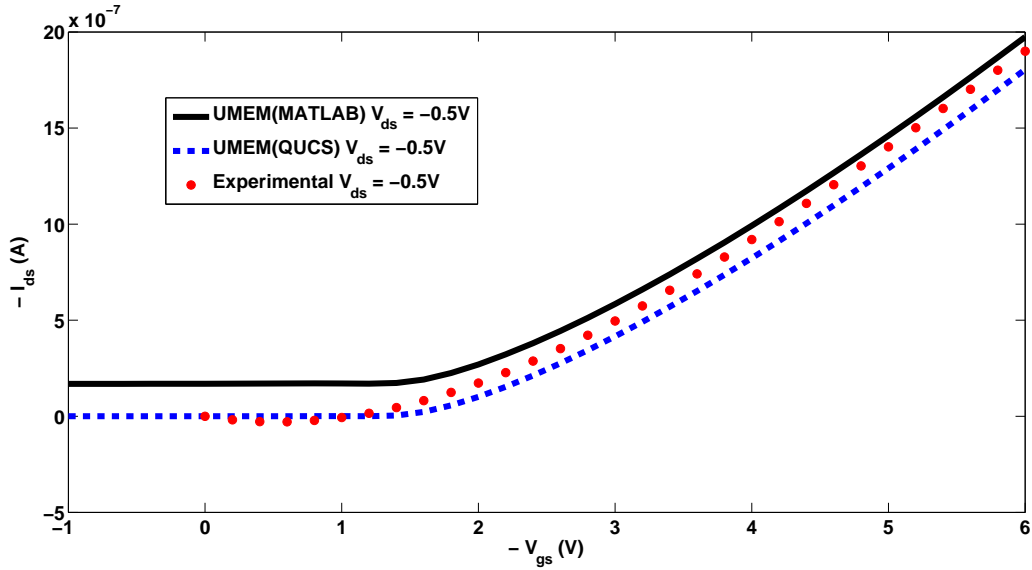


Figura 4.33: Curvas características de transferência para o TUD OTFT encontradas experimentalmente e modeladas pelo MATLAB e QUCS para $V_{ds} = -0.5V$.

(CINVESTAV OTFT no gráfico 4.32), as diferenças são muito pequenas. Além disso, como I_0 , I_{0S} e $\frac{V_{ds}}{CC}$ são constantes em uma curva de transferência, as transcondutâncias das curvas modeladas pelo QUCS e MATLAB apresentam o mesmo valor, já que a derivada desses termos é nula. A transcondutância para o gráfico 4.33 pode ser vista no gráfico 4.35.

Conclui-se nessa seção, então, que a aproximação da corrente de saída por uma única equação (correntes de fuga anuladas) é melhor para a maioria dos casos, pois como V_T é geralmente negativo (para os transistores tipo P), extrapolações para $|V_{gs}| < 0$ produzem erros consideráveis

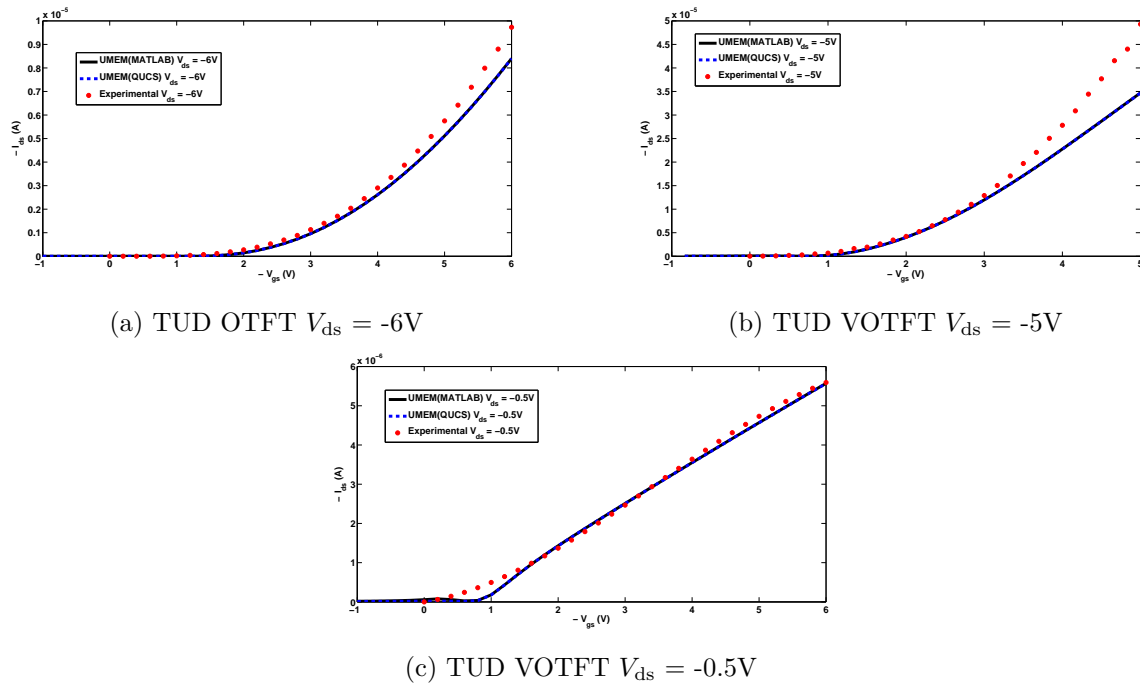


Figura 4.34: Curvas características de transferência encontradas experimentalmente e modeladas pelo MATLAB e QUCS.

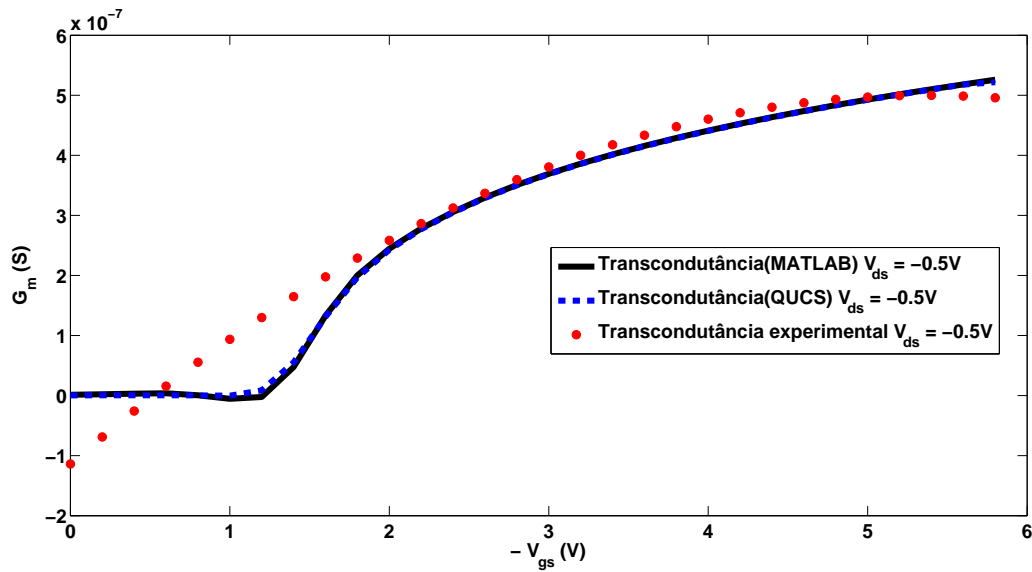


Figura 4.35: Curvas de transcondutâncias para o TUD OTFT encontradas experimentalmente e modeladas pelo MATLAB e QUCS para $V_{ds} = -0.5V$.

na estimação de I_0 . Mesmo que V_T seja positivo e a corrente modelada, para I_0 , $I_{0S} = 0$, se afaste levemente dos valores experimentais, as curvas possuirão as mesmas transcondutâncias e portanto, não afetarão as análises em aplicações envolvendo ampliações de pequenos sinais. Com isso, pode-se usar os parâmetros extraídos do UMEM em um simulador de circuitos, para o qual as correntes são calculadas através de uma única equação, independentemente das condições

de polarização. É importante destacar também que essa consideração de anular I_0 foi aplicada na seção anterior, já que V_T é negativo para os dois transistores, TUD VOTFT e TUD OTFT, com o objetivo de melhorar a aproximação com os dados experimentais e portanto a convergência dos parâmetros ajustáveis do modelo.

4.6 Aplicações do TUD VOTFT em topologias de circuito usando QUCS

Após o procedimento de otimização do UMEM e validação do código em Verilog A, foram utilizados os parâmetros da terceira simulação na tabela 4.4 da seção 4.3 para modelar o TUD VOTFT em topologias básicas de circuitos inversores amplificadores. Os parâmetros ajustáveis do modelo Q, S e DV escolhidos para modelar o transistor no QUCS foram aqueles obtidos através do método do menor valor máximo na tabela 4.6, para os quais as curvas de transcondutância foram mais satisfatórias. Para as topologias escolhidas, sabe-se que tanto os valores da corrente de polarização (controle da resistência de saída), quanto a transcondutância são importantes na determinação do ganho. No entanto, como os valores das correntes encontradas através dos dois métodos são aproximadamente iguais (exceto na curva de transferência para $V_{ds} = -0.5V$), a preferência foi dada aos parâmetros que resultaram em melhor transcondutância.

4.6.1 Amplificador Fonte comum

A implementação do amplificador fonte comum pode ser vista na figura 4.36. Na curva da característica estática dessa topologia, no gráfico 4.37, a tensão de saída máxima é atingida quando o transistor é cortado, o que ocorre para V_{gs} próximo à $V_T = 0.676V$, conforme esperado. Outro ponto observado foi o alto valor da resistência utilizada para se obter a excursão da tensão DC de saída mostrada. Isso pode ser explicado pelos altos valores de R_{on} nas diversas curvas saída (inclinações menores para $I_{ds} \times V_{ds}$), que demandam uma reta de carga com inclinação menor para se obter maiores excursões, interceptando essas curvas em valores de tensão V_{out} menores.

4.6.2 Amplificador inversor com carga ativa

A implementação do circuito correspondente a essa topologia pode ser vista no gráfico 4.38. É importante notar que se espera um ganho de tensão maior e uma excursão de tensão DC menor em relação ao circuito com o amplificador fonte comum. No entanto, observa-se no gráfico 4.39 que a inclinação da curva é semelhante a da topologia anterior, o que pode ser justificado pela alto valor da resistência utilizada na topologia fonte comum (igual à $100 \times 10^3 \Omega$), aumentando o ganho e portanto a inclinação da curva em 4.37. Nota-se também que o valor máximo de V_{out} atingido é superior ao esperado $V_{DD} - V_T = 6 - 0.676 = 5.324V$, mostrando que o transistor correspondente à carga ativa conduz significativamente na região sublimiar, o que pode ser explicado com base nos erros de interpolação dos dados experimentais da curva de transferência explicados anteriormente. Como o transistor correspondente à carga ativa sempre opera em saturação, quando baixos valores

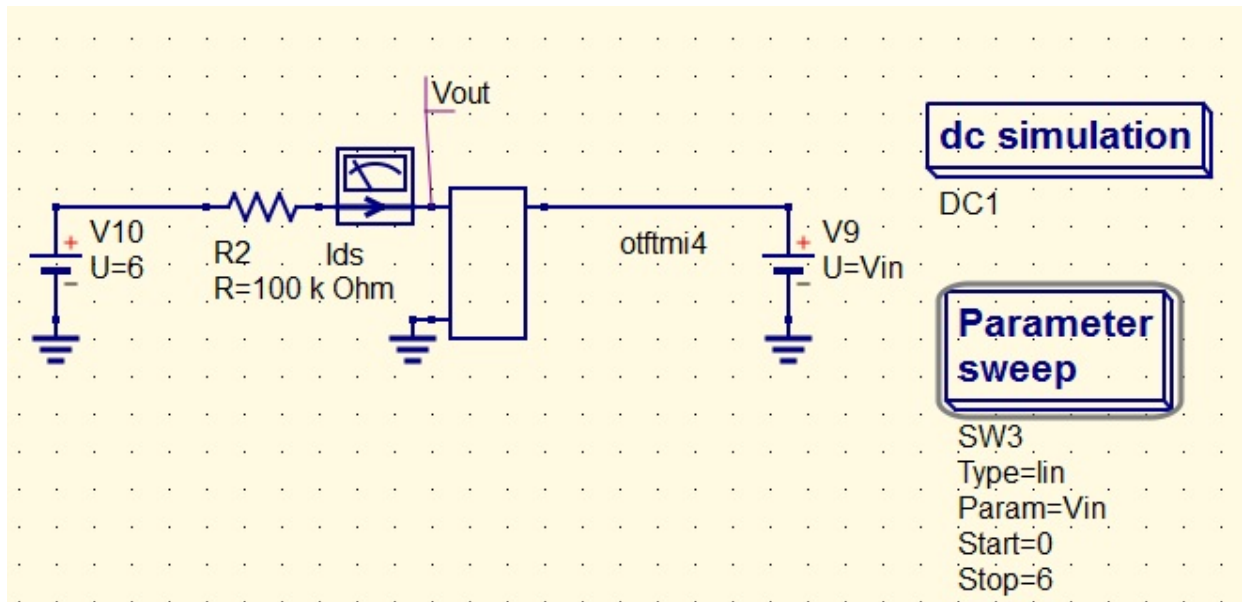


Figura 4.36: Amplificador fonte comum.

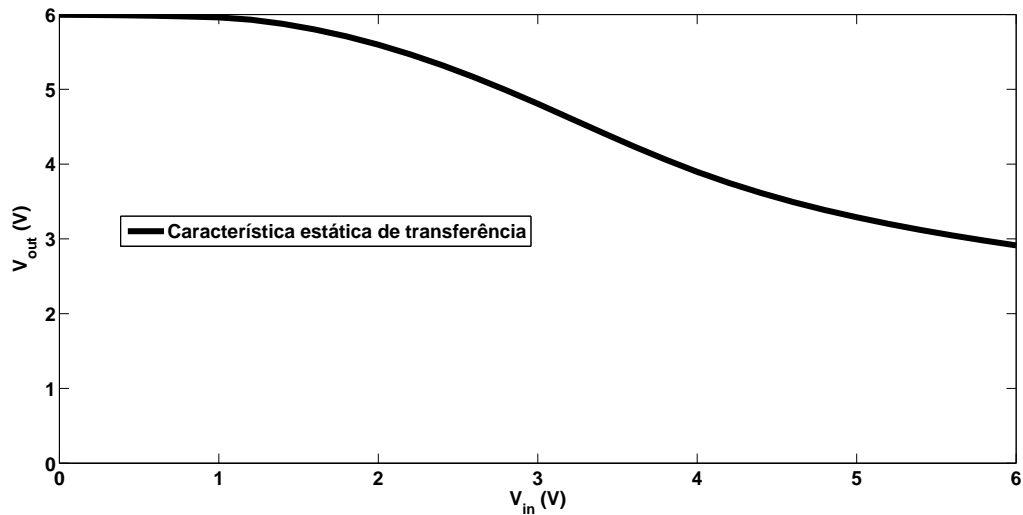


Figura 4.37: Característica estática de transferência de tensão do amplificador fonte comum.

de tensão são aplicados no gate do transistor tipo N, V_{out} aumenta, V_{sd} e V_{sg} do transistor P reduzem, levando-o a operar na região do gráfico 4.26b com pior aproximação dos dados experimentais (entre 0V e $-V_T$), onde a corrente deveria decrescer assintoticamente, mas decresce linearmente.

4.6.3 Amplificador inversor com fonte de corrente

Esse amplificador, cuja topologia é representada na figura 4.40, possui característica estática de transferência de tensão, no gráfico 4.41, que corresponde ao esperado. A distinção entre a inclinação desta curva e a anterior (carga ativa) é considerável. Nota-se que a excursão do sinal DC de saída também é maior, tanto através do limite superior ($V_{out,MAX}$), quanto inferior ($V_{out,MIN}$).

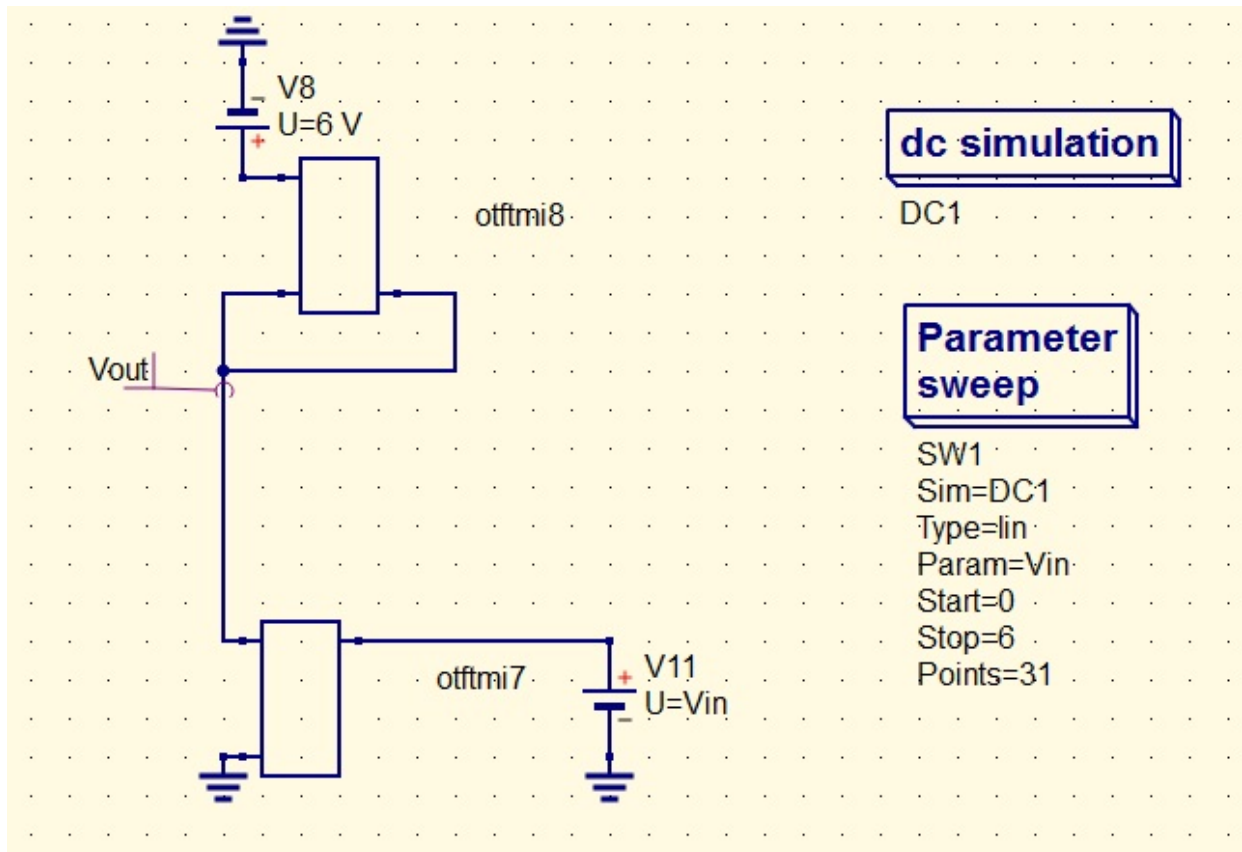


Figura 4.38: Amplificador inversor com carga ativa.

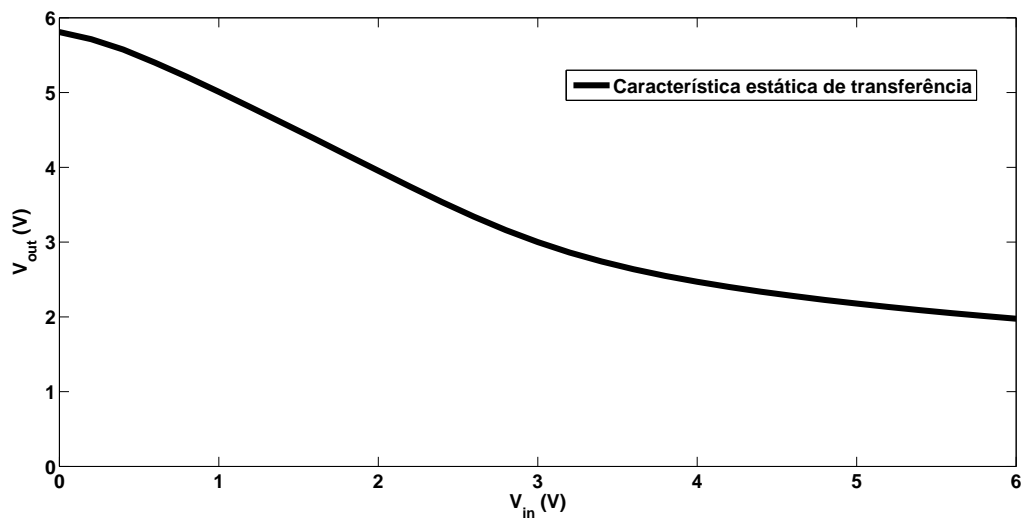


Figura 4.39: Característica estática de transferência de tensão do amplificador inversor com carga ativa.

Como os dois transistores usados possuem parâmetros extraídos de um OTFT tipo P, que possui mobilidade maior do que um OTFT tipo N (injeção de buracos é mais fácil em um semicondutor orgânico), é esperado uma inclinação menor da curva, no gráfico 4.41, caso um tipo N seja usado

como fonte comum. No entanto, isso pode ser compensado através do aumento da razão de aspecto ($\frac{W}{L}$) do transistor. Nota-se também que como V_{sg} é pequeno no transistor fonte de corrente, ele opera em saturação para uma maior excursão do sinal de saída V_{out} . Ou seja, entre $V_{out} < 2.676V$ e $V_{out} > V_{in} - 0.68$, ambos transistores estão saturados e a inclinação da curva é mais acentuada (maior ganho).

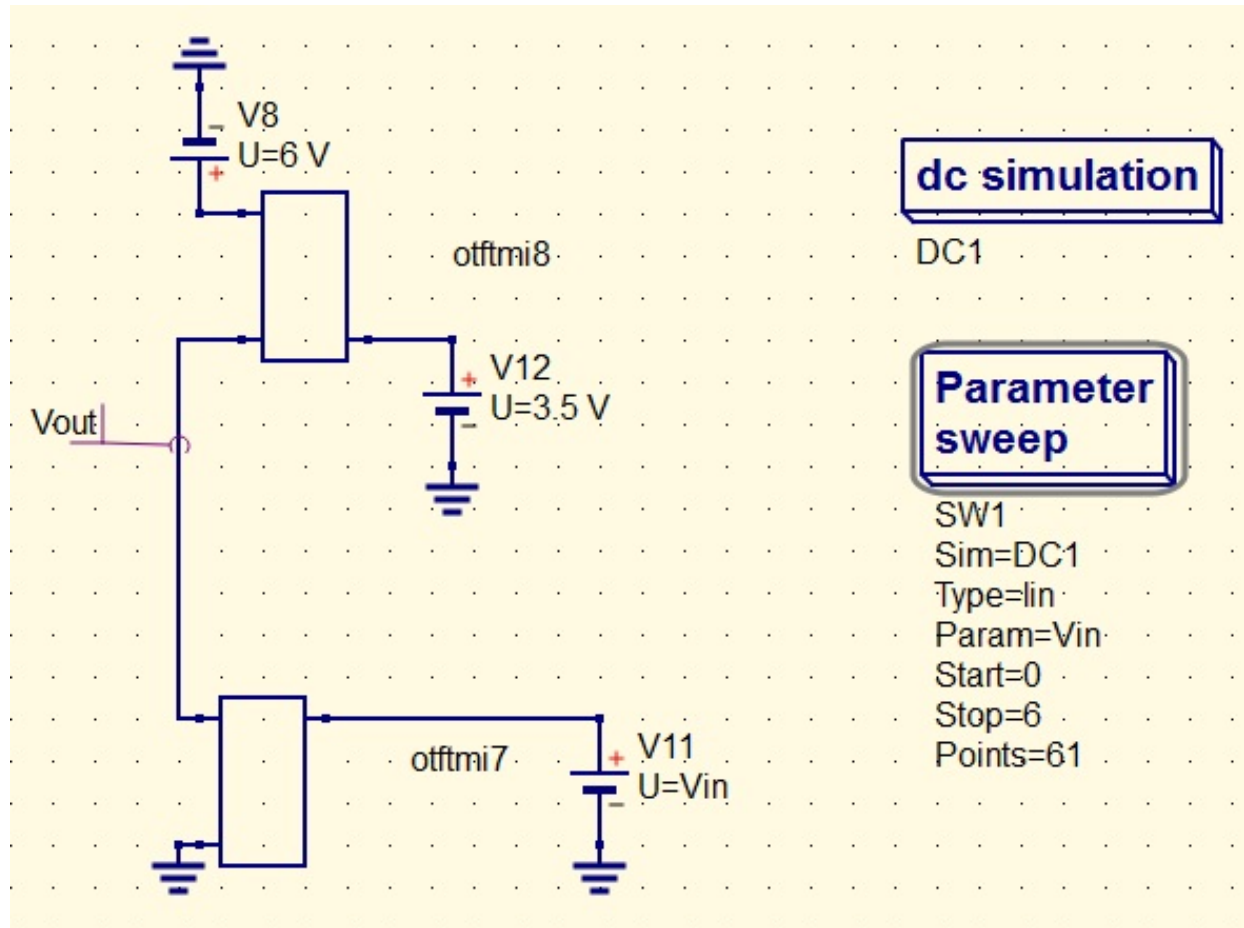


Figura 4.40: Amplificador inversor com fonte de corrente.

4.6.4 Amplificador inversor push-pull

Essa foi a ultima topologia testada e, de acordo com o esperado, obteve a maior excursão, variando entre 0 e V_{DD} , conforme pode ser visto no gráfico 4.43. No entanto, as diferenças entre as inclinações da curva de transferência estática desta topologia e da anterior não são tão claras (a inclinação do amplificador push-pull deve ser maior, pois possui maior ganho). No entanto, isso pode ser somente uma consequência da dificuldade em visualizar a região em que ambos transistores estão em saturação, a qual não pode ser controlada por uma fonte de tensão externa, como no amplificador fonte de corrente. Caso a razão de aspecto do transistor otftmi8 na figura 4.42 seja aumentada, a região em ambos transistores operam em saturação será expandida e o ganho desta topologia em relação a anterior se tornará mais visível.

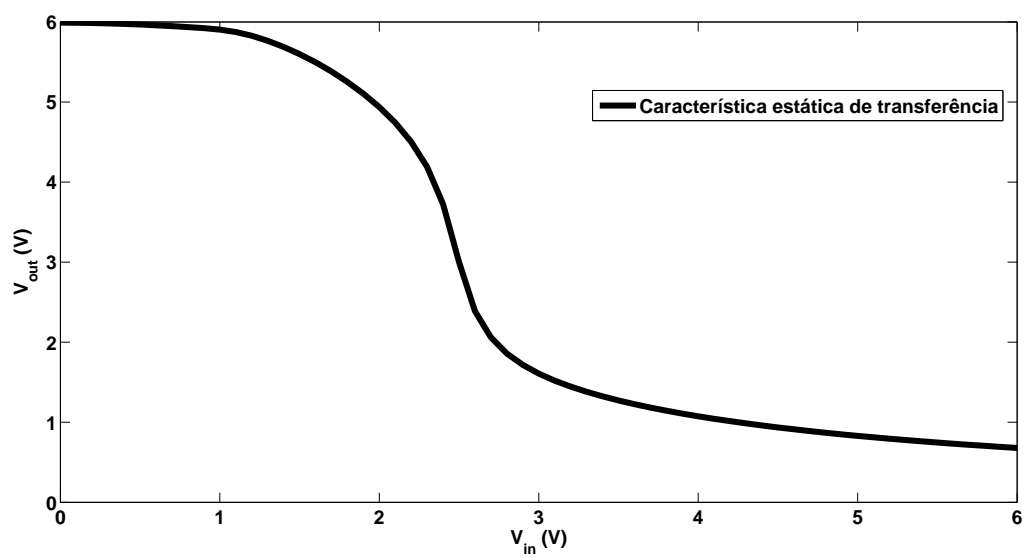


Figura 4.41: Característica estática de transferência de tensão do amplificador inversor com fonte de corrente.

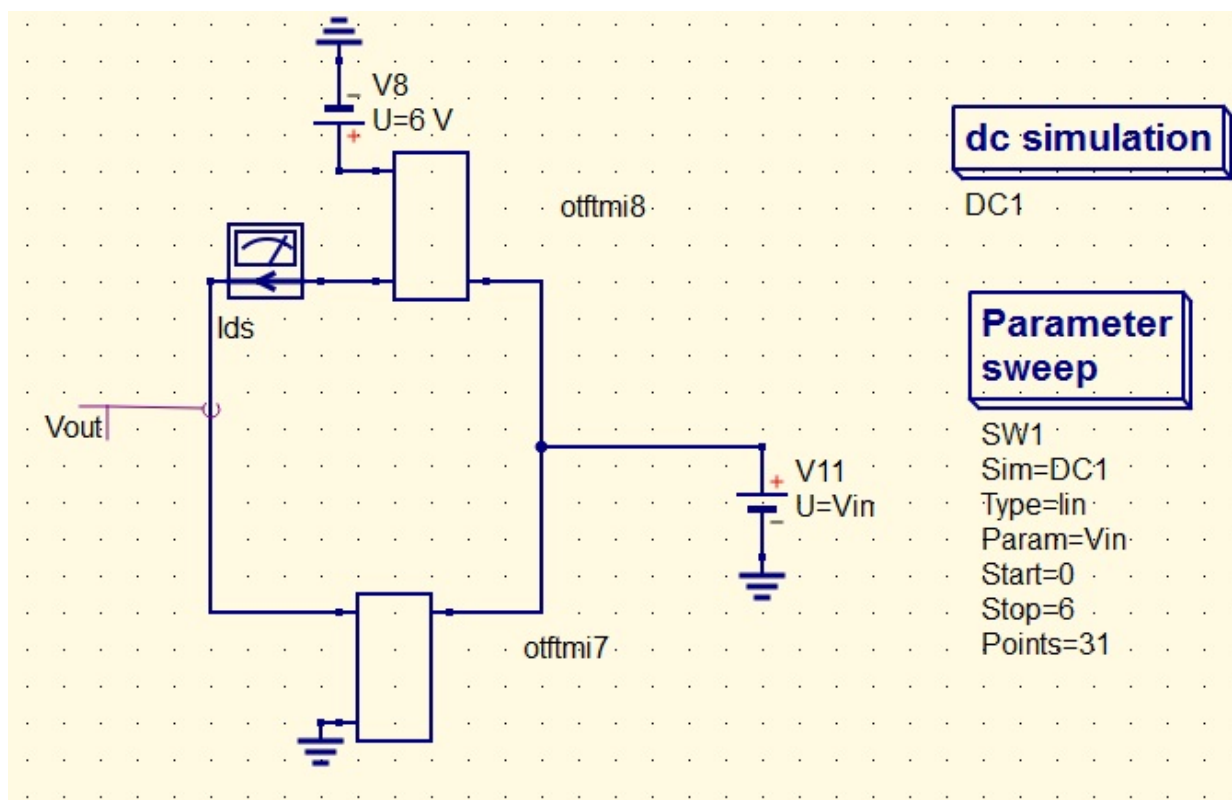


Figura 4.42: Amplificador inversor push-pull.

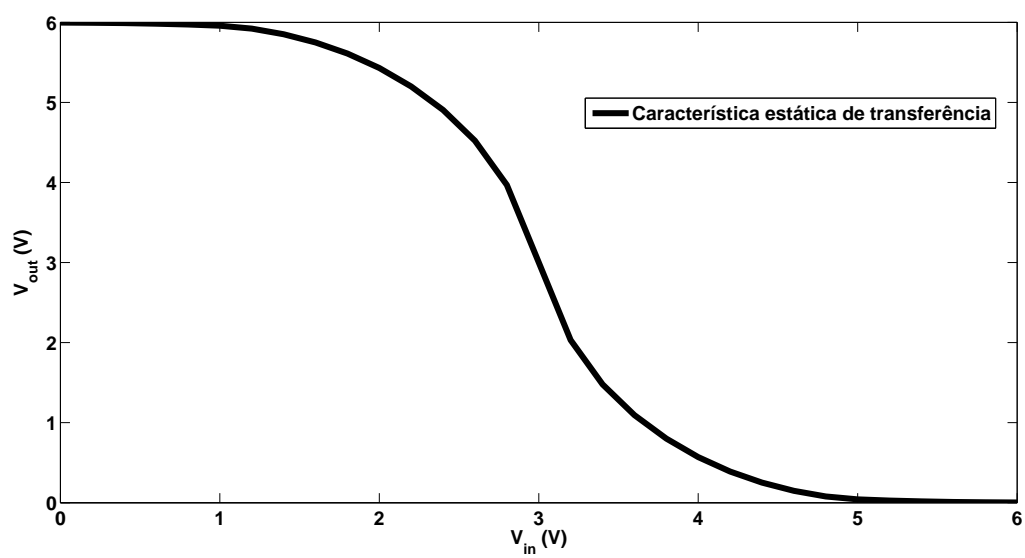


Figura 4.43: Característica estática de transferência de tensão do amplificador inversor push-pull.

Capítulo 5

Conclusões

Ao longo deste trabalho, foi possível executar todos os procedimentos propostos. Através da validação do método de extração do UMEM no MATLAB, assim como a implementação do modelo em Verilog A/QUCS, é possível usar as duas plataformas de forma integrada em simulações de circuitos a partir de curvas experimentais de quaisquer OFET's. Assim, uma infraestrutura de calibração e aplicação de um modelo para OFET's é obtida. Com a generalização do código no MATLAB, é possível aplicar o UMEM para outros OTFT's de maneira simples, alterando somente os parâmetros físicos do transistor e os vetores correspondentes às curvas de entrada. O número de pontos usados para a interpolação dos dados experimentais pode ser escolhido pelo usuário e a extração, assim como o ajuste, podem ser obtidos com uma boa precisão sem um esforço computacional maior.

O fluxograma mostrado na figura 3.1 do capítulo 3 pode ser utilizado como guia para demais aplicações dessas duas plataformas (MATLAB + QUCS) conjuntamente. Para obtenção de um ajuste satisfatório às curvas experimentais, utiliza-se as otimizações propostas na determinação dos parâmetros de ajuste da extração ($V_{H,min}$, $V_{H,max}$, $V_{IPC,min}$ e $V_{IPC,max}$) e ajustáveis do modelo (Q , DV e S). No entanto, uma estratégia adicional deve ser seguida. Como os módulos de extração e modelo estão integrados no MATLAB, demanda-se muito tempo para executar os dois ajustes simultaneamente, já que são realizadas 400 mil iterações por simulação durante a determinação de Q , S e DV . Logo, após a execução da primeira simulação, deve-se usar os parâmetros ajustáveis do modelo encontrados nas demais simulações (comentar linhas de código correspondentes ao algoritmo), determinando, assim, os parâmetros de ajuste da extração. Por fim, simula-se o algoritmo que determina Q , DV e S novamente, reduzindo o tempo total de simulação.

Através dos métodos de otimização propostos, extraiu-se os parâmetros do UMEM para duas tecnologias diferentes: TUD OTFT e TUD VOTFT. Os resultados mostraram que o ajuste do modelo às curvas experimentais depende de como estas são obtidas, principalmente a curva de transferência linear. O método que calcula os parâmetros ajustáveis do modelo através da menor média dos erros relativos tende a apresentar melhores resultados para uma curva experimental obtida com poucos erros devido a interpolações e flutuações. Enquanto a interpolação pode ser apropriadamente realizada com um número maior de medidas, as flutuações são praticamente

inevitáveis durante o experimento. Outra observação relevante é que erros de interpolação não somente alteram os valores de Q , Q_S , DV , DVS , S e SS , mas também afetam os parâmetros extraídos, através dos valores obtidos para a função $H(V_{gs})$. Assim, o ajuste é afetado pela determinação inadequada de todos os parâmetros. Possivelmente, se as medições da corrente para a curva de transferência linear fossem feitas com os mesmos pontos utilizados na determinação da curva de transferência de saturação, o ajuste poderia se aproximar do obtido para o CINVESTAV OTFT, para o qual a função $H(V_{gs})$ foi estimada corretamente, já que $V_T > 0$ nesse transistor.

Um estudo de caso com amplificadores inversores também foi realizado. Após a determinação dos parâmetros extraídos e ajustáveis do modelo, através das otimizações propostas, para um VOTFT, um transistor equivalente foi criado e simulado no QUCS. As topologias de circuito incluíram: Inversores com fonte comum (único transistor), carga ativa, fonte de corrente e push-pull. A simulação foi somente DC e as análises foram feitas a partir das curvas de transferência de tensão estática. Uma observação importante foi realizada para o amplificador inversor com carga ativa em relação ao efeito da interpolação inadequada das curvas experimentais. Foi notado que isso levou à condução durante o estado OFF (sublimar), o que poderia afetar o desempenho de circuitos que operam como chaves lógicas, por exemplo. Ainda assim, não foram realizadas análises AC, as quais poderiam mostrar como as limitações relacionadas a interpolações inadequadas ou flutuações afetam a operação em pequenos sinais.

5.1 Recomendações para trabalhos futuros

Com o proposito de verificar a validade da sugestão feita acima em relação ao número de medidas necessárias para determinação das curvas de transferência linear, pode-se utilizar um simulador computacional de OTFT's. Isso dispensaria as dificuldades, custos e tempo empregados para realização de experimentos. Recomenda-se que, para uma excursão de tensão semelhante à dos transistores usados nesse projeto (entre 0 e -6V), as medidas sejam realizadas com um passo de 0.2V entre elas para todas as curvas, incluindo a região sublimar, assim como foi feito na determinação das curvas experimentais de saída e transferência (saturação) dos transistores TUD VOTFT e TUD OTFT. Em relação a essas simulações, é recomendado também que se adquira os dados experimentais e verifique-se a validade do UMEM para um transistor tipo N.

Uma das limitações desse projeto é o uso de análises gráficas para extração de parâmetros do modelo, que são bastante vulneráveis à erros de acurácia. Esses poderiam ser minimizados através da determinação e comparação do desvio quadrático médio entre as curvas modelas e experimentais para diferentes intervalos dos parâmetros de ajuste da extração.

A separação entre os códigos correspondentes à extração e ao modelo do UMEM em diferentes módulos no MATLAB reduziria o tempo de simulação. Como a determinação dos parâmetros ajustáveis requer muitas iterações, as curvas I_{ds} poderiam ser analisadas inicialmente para $V_{gs} > V_T$ (sem considerar o produto com a tangente hiperbólica), determinando em poucas simulações os parâmetros $V_{H,min}$, $V_{H,max}$, $V_{IPC,min}$ e $V_{IPC,max}$. Em seguida, através do módulo correspondente ao modelo UMEM, encontra-se os parâmetros ajustáveis Q , DV e S .

Em relação a determinação dos parâmetros ajustáveis, a curva de transferência de saturação poderia ser utilizada na determinação do menor erro relativo, já que a mesma foi interpolada apropriadamente. No entanto, isso não afetaria a extração e os valores da função $H(V_{gs})$ ainda precisariam ser corrigidos através das medições sugeridas anteriormente para a curva de transferência linear.

Outra sugestão é a adaptação do método de extração de parâmetros do UMEM à outros modelos, tais como Marinov et al [5] e Li et al [6]. Como a extração dos parâmetros é obtida de forma prática através da equação que define a corrente, esta poderia ser modificada de acordo com novas considerações, em relação a mobilidade (variável com a tensão $V(x)$ ao longo do canal), por exemplo.

Outra possibilidade para trabalhos futuros é a inclusão de capacitâncias parasitas no modelo, tornando possível a análise AC, para altas frequências, das topologias usadas neste projeto. Em seguida, circuitos com osciladores poderiam ser feitos, afim de simular a operação do VOTFT em sistemas RFID, uma das aplicações principais para transistores orgânicos.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] LIMA, A. R. de A. *Complemento de um simulador geral de circuitos (QUCS) com um modelo e método de extração unificados (UMEM)*. Dezembro 2015. Universidade de Brasília.
- [2] KLAUK, H. *Organic Electronics: Materials, Manufacturing and applications*. [S.l.]: Wiley-VCH, 2006.
- [3] LÜSSEM, B. et al. Vertical organic transistors. *Journal of Physics: Condensed Matter*, v. 27, Outubro 2015.
- [4] KIM, C.-H.; BONNASSIEUX, Y.; HOROWITZ, G. Compact dc modeling of organic field-effect transistors: Review and perspectives. *IEEE TRANSACTIONS ON ELECTRON DEVICES*, v. 61, n. 2, p. 278–287, Fevereiro 2014.
- [5] MARINOV, O. et al. Organic thinfilm transistors: Part i - compact dc modeling. *IEEE Trans. Electron. Devices*, v. 56, n. 12, p. 2952 – 2961, Dezembro 2009.
- [6] LI, L. et al. A compact model for polycrystalline pentacene thin-film transistor. *J. Appl. Phys.*, v. 107, n. 2, p. 024519 – 024519–3, Janeiro 2010.
- [7] ESTRADA, M. et al. Accurate modeling and parameter extraction method for organic tfts. *Solid-State Electronics*, v. 49, p. 1009–1016, 2005.
- [8] CERDEIRA, A. et al. New procedure for the extraction of basic a-si:h tft model parameters in the linear and saturation regions. *Solid-State Electron*, v. 45, p. 1077–1080, Fevereiro 2001.
- [9] CERDEIRA, A. et al. Modeling the subthreshold region of otfts. *8th International Conference on Electrical Engineering, Computing Science and Automatic Control*, 2011.
- [10] ALLEN, P. E.; HOLBERG, D. R. *CMOS Analog Circuit Design*. [S.l.]: Oxford University Press, 2002.
- [11] SEDRA, A. S.; SMITH, K. C. *Microeletrônica*. [S.l.]: Pearson, 2007.
- [12] H.KLEEMANN et al. High-performance vertical organic transistors. *Small*, v. 9, n. 21, p. 3670–3677, 2013.

ANEXOS

I. CÓDIGO NO MATLAB

I.1 Módulo da definição dos parâmetros físicos

```
% This files contains all the parameters used during the OTFT Modeling
% electron devices project.

% 1. Physical constants
q          = 1.6e-19;      % Magnitude electron's charge [C]
kb         = 8.62e-5;      % Boltzmann's constant      [eV/K]
e0         = 8.86e-14;     % Vacuum's permittivity      [F/cm]

% 2. Input Data

% 2.1 Measurement temperature
T          = 300;          % Temperature                [K]

% 2.2 Dimmensions
W          = 0.1000;       % Width                [cm]
L          = 0.0050;       % Length                [cm]
Lovr       = 15e-4;        % Length overlap        [cm]
di         = 3.0000e-06;    % Insulator thickness    [cm]

% 3. Electrical parameters (INSERT ABSOLUTE VOLTAGES VALUES)

eir        = 2.6000;       % Relative dielectric constant [F*cm]
% of the dielectric layer
esr        = 9.4;          % Relative dielectric constant [F*cm]
% of the semiconductor layer
Stype      = 2;            % Semiconductor type N type=1
% p type=2
Nc         = 2e21;          % Density of states at HOMO    [cm-3]
Nv         = 2e21;          % Density of states at LUMO    [cm-3]

Vd1        = 0.5;          % Drain voltage from linear    [V]
% transfer characteristic

Vgsmax     = 6;            % Maximum gate voltage.        [V]
% Linear TF measured

VgsmaxS    = 6;            % Maximum gate voltage.        [V]
% Saturation TF measured

Vd2        = 6;            % Drain voltage in saturation  [V]

Vdmax      = 6;            % Maximum drain voltage        [V]
% at output

% 3.1 Gate voltages for output characteristics(INSERT ABSOLUTE VOLTAGES VALUES) [V]

Vgs1       = 0;
Vgs2       = 2;
Vgs3       = 4;
Vgs4       = 6;
```

```

%Vgs5          = ;
%Vgs6          = ;
Vgs_out = [Vgs1, Vgs2, Vgs3, Vgs4]; % Vector of Vgs values created to be used in calculations of Vds in output curves

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 4. Calculation of parameters
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Vth          = kb*T;          % Thermal voltage          [V]
ei           = eir*e0;        % Permittivity interface
% semiconductor-dielectric    [V/cm-s]
Ci           = ei/di;         % Capacitance interface
% semiconductor-dielectric    [C]
K            = (W/L)*Ci;      % Multiplicative constance [C]

% INPUT LEAKAGE CURRENT - SATURATION REGION (INSERT ABSOLUTE CURRENT VALUE)

IoS          = 6.6589e-10;

% PARAMETERS FOR H AND IPS1 FUNCTIONS ADJUSTMENT

Vgs_Hmin      = (3/6); % Specify the minimum voltage ratio (Vgs) for the H function      (LTC)
Vgs_Hmax      = (5/6); % Specify the maximum voltage ratio (Vgs) for the H function      (LTC)
Vgs_ips1_min  = (4/6); % Specify the minimum voltage ratio (Vgs) for the ips1 function   (STC)
Vgs_ips1_max  = (6/6); % Specify the maximum voltage ratio (Vgs) for the ips1 function   (STC)

% PARAMETERS FOR INTERPOLATION

N = 31; % Number of point desired to interpolate data. " Vdmax/(N-1) " and " VgsmaxS/(N-1) " and " Vgsmax/(N-1) "
% must be numbers with at the most 3 decimal places.

```

I.2 Módulo do carregamento e armazenamento dos dados correspondentes às curvas experimentais

% This files contains all the functions used to extract the data to model
% the OTFT.

```
('ParameterDef_OFET');
```

```
% 1. Reading MS Excel files Vgs x Id_linear
```

```

filename_Vg      = 'ID-VG.xls';
sheet_Vg         = 2;

xlRange_Vg_LTC   = 'A1:A4';
xlRange_Vg_LTC_idlreal = 'B1:B4';

```

```

Vg_LTC = abs(xlsread(filename_Vg, sheet_Vg, xlRange_Vg_LTC));
Vg_LTC_idlreal = abs(xlsread(filename_Vg, sheet_Vg, xlRange_Vg_LTC_idlreal));

% 2. Reading MS Excel files Vgs SAT vs Ids SAT

filename_VgId_SAT = 'ID-VG.xls';
sheet_VgId_SAT = 1;

xlRange_Vg_SAT = 'A1:A61';
xlRange_Id_SAT = 'B1:B61';

VgSAT = abs(xlsread(filename_VgId_SAT, sheet_VgId_SAT, xlRange_Vg_SAT));
IdSAT = abs(xlsread(filename_VgId_SAT, sheet_VgId_SAT, xlRange_Id_SAT));

% 3. Reading MS Excel files Vds

filename_Vd = 'ID-VD.xls';
sheet_Vd = 1;

xlRange_Vd = 'A2:A62';
xlRange_Vd_Vgs1r = 'B2:B62';
xlRange_Vd_Vgs2r = 'C2:C62';
xlRange_Vd_Vgs3r = 'D2:D62';
xlRange_Vd_Vgs4r = 'E2:E62';
%xlRange_Vd_Vgs5r = 'F2:F62';
%xlRange_Vd_Vgs6r = 'G2:G62';
%Add more rows if needed

Vd = xlsread(filename_Vd, sheet_Vd, xlRange_Vd);
Vd_Vgs1r = xlsread(filename_Vd, sheet_Vd, xlRange_Vd_Vgs1r);
Vd_Vgs2r = xlsread(filename_Vd, sheet_Vd, xlRange_Vd_Vgs2r);
Vd_Vgs3r = xlsread(filename_Vd, sheet_Vd, xlRange_Vd_Vgs3r);
Vd_Vgs4r = xlsread(filename_Vd, sheet_Vd, xlRange_Vd_Vgs4r);
%Vd_Vgs5r = xlsread(filename_Vd, sheet_Vd, xlRange_Vd_Vgs5r);
%Vd_Vgs6r = xlsread(filename_Vd, sheet_Vd, xlRange_Vd_Vgs6r);
%Add more rows if needed

Vd_idCCr = abs(Vd_Vgs1r);
Vd = abs(Vd);

Vd_idr = zeros(length(Vgs_out), length(Vd_Vgs2r));
Vd_idr(1,:) = abs(Vd_Vgs1r);
Vd_idr(2,:) = abs(Vd_Vgs2r);
Vd_idr(3,:) = abs(Vd_Vgs3r);
Vd_idr(4,:) = abs(Vd_Vgs4r);
%Vd_idr(5,:) = abs(Vd_Vgs5r);
%Vd_idr(6,:) = abs(Vd_Vgs6r);
%Add more rows if needed

```

I.3 Módulo com as interpolações

```

% This files contains all the interpolation used in the modeling of the
% OTFT.

('Dataextract_OFET');

% 1. Interpolation characteristic curves

% 1.1 Linear transfer curves: Cubic spline interpolation & Interpolation 1D

xx_spline          = 0:(max(Vg_LTC)/(N-1)):max(Vg_LTC);
idl1kreal          = spline(Vg_LTC, Vg_LTC_idlreal,xx_spline);

% 1.2 Saturation transfer curves: Cubic spline interpolation & Interpolation 1D

xx_spline_sat      = 0:(max(VgSAT)/(N-1)):max(VgSAT);
iksatr             = spline(VgSAT, IdSAT, xx_spline_sat);

% 1.3 Measured output curves: Cubic spline interpolation Sxr &
% Interpolation 1D ixreal

xx_spline_measured = 0:(Vdmax/(N-1)):Vdmax;

iCCreal            = spline(Vd, Vd_idCCr,xx_spline_measured);

ireal              = zeros(length(Vgs_out), length(xx_spline_measured));

for Vgs = 1:length(Vgs_out)
ireal(Vgs,:)       = spline(Vd, Vd_idr(Vgs,:),(xx_spline_measured)');
end

% 2. Calculation of the correction factor CC from the curve CC in the
% sub-threshold regime

xx_interp_slope    = 0:0.001:Vdmax;
iCCr               = interp1(xx_spline_measured, iCCreal, xx_interp_slope, 'linear');

VCCmin             = round(Vdmax - 2);
aCC1               = (Vdmax - VCCmin)*1000;
k11                = zeros (1,aCC1+1);
l11                = zeros (1,aCC1+1);

for iCC1 = 1:(aCC1+1)
k11(iCC1) = VCCmin + iCC1*0.001 - 0.001;
end

for iCC1 = 1:(aCC1+1)
l11(iCC1) = iCCr((VCCmin*1000)+ iCC1); % iCC1 - 1;
end

PCC                = (l11(aCC1+1)-l11(1))/(k11(aCC1+1)-k11(1));
CC                 = 1/PCC;

% Consider CC infinite when measured current corresponds to noise (For Vgs = 0V)

% 2.1 Correction linear transfer curve

```

```

idlcor = zeros(1,length(Vg_LTC_idlreal));

for i = 1:length(Vg_LTC_idlreal)
idlcor(i) = Vg_LTC_idlreal(i)-(Vd1/CC);
end

% Interpolation of the corrected linear transfer curve: spline &
% interp1D

xx_spline_lincor = 0:(max(Vg_LTC)/(N-1)):max(Vg_LTC); %-15:1:50;
idllkcor = spline(Vg_LTC, idlcor, xx_spline_lincor);

% 2.2 Correction saturation transfer curve

idsatcor = zeros(1,length(IdSAT));

for i = 1:length(IdSAT)
idsatcor(i) = IdSAT(i)-(Vd2/(CC));
end

% Interpolation of the corrected saturation transfer curve: spline &
% interp1D

xx_spline_satcor = 0:(max(VgSAT)/(N-1)):max(VgSAT);
iksatcor = spline(VgSAT, idsatcor,xx_spline_satcor);

% Interpolation iksatcor

xx_interp_sat_p = 0:0.001:max(VgSAT);
iksatcor_p = interp1(xx_spline_satcor, iksatcor, xx_interp_sat_p, 'linear');

% 2.3 Correction of the output characteristic curves

idcor = zeros(length(Vgs_out),length(Vd_Vgs1r));
idCCcor = zeros(1,length(Vd_idCCr));

for Vgs = 1:length(Vgs_out)
for Vds = 1:length(Vd_Vgs1r)
idcor(Vgs, Vds) = Vd_idr(Vgs,Vds)-((Vd(Vds))/(CC));
end
end

for Vds = 1:length(Vd_idCCr)
idCCcor(Vds) = Vd_idCCr(Vds) - ((Vd(Vds))/(CC));
end

% Spline interpolation of the output characteristic curves

xx_spline_measured_cor = 0:(Vdmax/(N-1)):Vdmax;
icor = zeros(length(Vgs_out), length(xx_spline_measured_cor));

for Vgs = 1:length(Vgs_out)

```

```

icor(Vgs,:) = spline(Vd, idcor(Vgs,:),xx_spline_measured_cor);
end

iCCcor = spline(Vd, idCCcor, xx_spline_measured_cor);

% 3. H1 Auxiliary function: parameter calculation

n1 = 0:0.001:max(Vg_LTC);
xx_interp_lincor = 0:(max(Vg_LTC)/(N-1)):max(Vg_LTC);
idllkcor_p = interp1(xx_interp_lincor, idllkcor, n1, 'linear');

% 4. Current to calculate parameters: m and lambda

% The algorithm below finds the highest output curve which corresponds at the most to...
% the maximum transfer curve voltage Vgs measured.

xx_interp_measured_cor_p = 0:0.001:Vdmax;
Vector_positions = find(abs(min(abs(VgsmaxS - Vgs_out)) - abs(VgsmaxS - Vgs_out)) < 1e-10);
Vector_Position = min(Vector_positions);
imaxcor_p = interp1(xx_spline_measured_cor, icor(Vector_Position,:), ...
    xx_interp_measured_cor_p,'linear');
imaxcor = icor(Vector_Position,:);

% 5. Measured data to be compared to corrected modelled curves (output characteristics)

w1 = 0:(Vdmax/(N-1)):Vdmax;

ireal_inter = zeros(length(Vgs_out), length(w1));

for Vgs = 1:length(Vgs_out)
    ireal_inter(Vgs,:) = interp1(xx_spline_measured, ireal(Vgs,:), w1, 'linear');
end

iCCreal_inter = ireal_inter(1,:);

% 7. Estrada's compact model

idc_Vd1_interpolant = 0:(Vgsmax/(N-1)):Vgsmax;
idc_Vd1_inter_ISc = 0:0.001:Vgsmax;

idc_Vd2_interpolant = 0:(VgsmaxS/(N-1)):VgsmaxS;
idc_Vd2_inter_ISc = 0:0.001:VgsmaxS;

Vgs_Vd2 = 0:(VgsmaxS/(N-1)):VgsmaxS;

xs3 = 1:1:(N+5);
xs3 = (xs3-6)*(VgsmaxS/(N-1));

Vgs_Vd1 = 0:(Vgsmax/(N-1)):Vgsmax;

xs1 = 1:1:(N+5);
xs1 = (xs1-6)*(Vgsmax/(N-1));

```


I.4 Módulo com a implementação do UMEM (Extração + Modelo)

```
% This files contains all the functions used to implement the Estrada's  
% current compact model the OTFT.
```

```
('Interpolation_OFET');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 1. ABOVE THRESHOLD REGIME %%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Step 1: Calculation H1, gamma and Threshold voltage Vt
```

```
H1 = zeros(1,length(idllkcor_p));  
Vgsmin = (Vgs_Hmin)*Vgsmax;  
Vmaxcal = (Vgs_Hmax)*Vgsmax;  
a = round((Vmaxcal - Vgsmin)*1000);  
k1 = zeros(1,a+1);  
l1 = zeros(1,a+1);
```

```
for i1 = 1:a+1  
k1(i1) = Vgsmin + (0.001*i1)-0.001;  
end
```

```
% 3.1 Trapezoidal integration approach to calculate H1
```

```
for n2 = 1:length(idllkcor_p)  
cal = cumtrapz(n1,idllkcor_p);  
H1(1,n2) = cal(n2)/idllkcor_p(n2);  
end
```

```
for i1 = 1:a+1  
l1(i1) = H1(1, round(Vgsmin*1000)+i1);  
end
```

```
% Modification made above in order to account the fasct  
%that the first index corresponds to 0 V.
```

```
% 3.2 Threshold voltage Vt
```

```

Coefficients_Vt          = zeros(2,1);
Coefficients_Vt(:,1)    = polyfit(k1, l1, 1);
P_Vt                     = Coefficients_Vt(1,1);
inter_Vt                 = Coefficients_Vt(2,1);

Vt                       = -(inter_Vt)/P_Vt;

% Modification made to calculate slope and intercep of adjsusting function
%with higher precision

HH1                      = zeros(1,length(idllkcor_p));

for j = 1:length(n1)
HH1(j)                   = inter_Vt + P_Vt*((j-1)/1000);
end

% H function plot below

figure(4)
plot(n1, HH1, '-b', n1, H1, '--k', 'LineWidth', 5);
legend('HH1', 'H1');
ylabel('H', 'FontSize',16, 'Fontweight', 'b');
if Stype > 1
xlabel('- V_g_s (V)', 'FontSize',16, 'Fontweight', 'b');
else
xlabel('V_g_s (V)', 'FontSize',16, 'Fontweight', 'b');
end
set(gca, 'FontSize',16, 'Fontweight', 'b');

ii1                      = zeros(1,length(idllkcor_p));
iA1                      = zeros(1,length(a+1));

% Correction for the organic semiconductor type for the threshold voltage

if (Stype > 1)
VT = -Vt;
else
VT = Vt;
end

% 3.3 Gamma

gamma = (1/P_Vt)-2;

for n2 = 1:length(idllkcor_p)
ii1(1,n2) = idllkcor_p(n2)^(1/(1+gamma));
end

for i1 = 1:a+1
iA1(1,i1) = ii1(1, round(Vgsmin*1000)+i1);
end

```

```

% Step 4: calculation PA, slope for the linear region and Vaa, fitting
% parameter

% 4.1 Calculation PA1

Coefficients_Vaa      = zeros(2,1);
Coefficients_Vaa(:,1) = polyfit(k1,iA1,1);

P_Vaa                 = Coefficients_Vaa(1,1);

PA1                    = P_Vaa;

% 4.2 Calculation Vaa

Vaa                    = ((K*Vd1)/((PA1)^(1+gamma)))^(1/gamma);

% Step 5: Calculation of the mobility for low electric fields

ufet0                  = 1/(Vaa^gamma);

% Step 6: Model for mobility
ufet                    = zeros(1,length(idllkreal));
for n3 = 1:length(idllkreal)
ufet(n3) = ufet0*((xx_spline(n3) - Vt)^gamma);
end

% Step 7: Calculation of contact resistance R

R1                      = (Vd1/(idllkcor_p((Vgsmax*1000)+1))) - (1/(K*ufet0*(Vgsmax-Vt)^(gamma+1)));
if (R1 > 0)
R = R1;
else
R = 0;
end

% Step 8: Calculation of PS: slope of the saturation region

iiks1                   = zeros(1,length(iksatscor_p));
ipc                     = zeros(1,length(iksatscor_p));
VgsSmin                 = (Vgs_ips1_min)*VgsmaxS;
VmaxcalS                = (Vgs_ips1_max)*VgsmaxS;
aS                      = round((VmaxcalS - VgsSmin)*1000);
ks1                     = zeros(1,aS+1);
ips1                    = zeros(1,aS+1);

for i1 = 1:aS+1
ks1(i1) = VgsSmin + (0.001*i1)-0.001;
end

for n4 = 1:length(iksatscor_p)
iiks1(n4) = mpower(iksatscor_p(n4), 1/(2+gamma));
end

for i1 = 1:aS+1

```

```

ips1(i1) = iiks1(round(VgsSmin*1000)+i1);
end

Coefficients_alphas          = polyfit(ks1,ips1,1);

P_alphas                     = Coefficients_alphas(1);

Ps                           = P_alphas;

% Used for comparison with iiks11

for x = 1:length(iksatcor_p)
ipc(x) = Ps*((x-1)*0.001)-Vt;
end

% Plot below of IIKS1 and IPC adjusting functions

figure(5)
plot(xx_interp_sat_p, real(ipc), '--k', xx_interp_sat_p, real(iiks1), '-b', 'LineWidth', 5);
legend('IPC','IIKS1');
if Stype > 1
xlabel('- V_g_s (V)', 'FontSize',16, 'Fontweight', 'b');
else
xlabel('V_g_s (V)', 'FontSize',16, 'Fontweight', 'b');
end
set(gca, 'FontSize',16, 'Fontweight', 'b');

% Step 9: Calculation of saturation voltage VDssat (pinch-off)

alphas                       = ((Vaa^gamma)/K)*(Ps^(2+gamma))*sqrt(2);
VgS                           = Vgs_out(1, Vector_Position);
V2                             = Vdmax;
VDsat                         = alphas*(VgS - Vt);

% Step 10: Calculation of m

iDsat                         = imaxcor_p((round(VDssat*1000))+1);

i2                             = imaxcor(length(imaxcor)); % Current value taken from last measurement of choosen
% output characteristic
dm                             = K*ufet0*(((VDsat/alphas)^(1+gamma))*(VDsat/iDsat))...
/(1+(R*(K*ufet0*((VDsat/alphas)^(1+gamma)))));
m                             = log(2)/log(dm);

% Step 11: Calculation of lambda

dlambda                       = (K*ufet0*((VDsat/alphas)^(1+gamma)))...
/((1+(R*(K*ufet0*((VDsat/alphas)^(1+gamma)))))*((1+((V2/VDsat)^m))^(1/m)));
lambda                       = ((i2/(V2^2))/(dlambda)) - (1/V2);

% Step 12: Calculation of Estrada's compact model for current

Vgsz                         = length(Vgs_out);
Vdsat_lowest                 = 0.751*alphas*(Vgs_out(2) - Vt);
% Estimation of lowest VdsSAT multiplied by a facto 0.751 in order to plot linear OC (if required)
Vdsz                         = (((round(Vdsat_lowest/(Vdmax/(N-1)))) + 1);
idc                         = zeros(Vgsz,N);
idc_Vgs                     = zeros(Vgsz, N);

```

```

idc_Vgs_lm = zeros(Vgsz, Vdsz);
idc_lm = zeros(Vgsz, Vdsz);
icor_lm = zeros(Vgsz, Vdsz);
idc_Vd1 = zeros(1, N);
idc_Vd2 = zeros(1, N);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ABOVE THRESHOLD ESTRADA'S COMPACT MODEL FOR CURRENT %%%%%%%%%%%%%%%
% 12.1. Output characteristic curves: linear & saturation %%%%%%%%%%%
% 12.2. Output characteristic curves in the linear region %%%%%%%%%%%
% 12.3 Linear transfer characteristic curve %%%%%%%%%%%%%%%
% 12.4 Saturation transfer characteristic curve %%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% 12.1 MODELED output characteristic curve in linear and saturation regions
for Vgs = 1:Vgsz
for Vds = 1:1:N;
idc(Vgs, Vds) = ((K*ufet0)/(1+(R*(K*ufet0)*((Vgs_out(Vgs))-Vt)^(1+gamma))))...
*(((Vgs_out(Vgs))-Vt)^(1+gamma))*(((Vds-1)/(N-1))*Vdmax)*(1+lambda*(((Vds-1)/(N-1))*Vdmax)))...
/((1+(((Vds-1)/(N-1))*Vdmax)/(alphas*((Vgs_out(Vgs))-Vt)))^m)^(1/m));
end
end

w2 = 0:((Vdmax)/(N-1)):Vdmax;

for Vgs = 1:Vgsz
for Vds = 1:1:N;
idc_Vgs(Vgs, Vds) = idc(Vgs, Vds);
end
end

% 12.2 MODELED output characteristic curve in linear region

for Vgs = 1:Vgsz
for Vds = 1:Vdsz
idc_lm(Vgs, Vds) = ((K*ufet0)/(1+(R*(K*ufet0)*((Vgs_out(Vgs))-Vt)^(1+gamma))))*...
(((Vgs_out(Vgs))-Vt)^(1+gamma))*(((Vds-1)/(N-1))*Vdmax)*(1+lambda*(((Vds-1)/(N-1))*Vdmax)))...
/((1+(((Vds-1)/(N-1))*Vdmax)/(alphas*((Vgs_out(Vgs))-Vt)))^m)^(1/m));
end
end

w0 = 0:((Vdmax)/(N-1)):((Vdsz-1)/(N-1))*Vdmax;

for Vgs = 1:Vgsz
for Vds = 1:Vdsz
idc_Vgs_lm(Vgs, Vds) = idc_lm(Vgs, Vds);
end
end

% Corrected output characteristic curves for comparison to the modeled
% characteristic curves.

for Vgs = 1:Vgsz
for Vds = 1:Vdsz
icor_lm(Vgs, Vds) = icor(Vgs, Vds);
end
end

%12.3 MODELED Linear transfer characteristic curve

```

```

for Vgs = 1:1:N
idc_Vd1(Vgs) = ((K*ufet0)/(1+(R*(K*ufet0)*(((Vgs-1)/(N-1))*Vgsmax)-Vt)^(1+gamma))))...
*(((Vgs-1)/(N-1))*Vgsmax)-Vt)^(1+gamma))*(Vd1)*(1+lambda*(Vd1)))...
/((1+((Vd1)/(alphas*(((Vgs-1)/(N-1))*Vgsmax)-Vt)))^m)^(1/m));
end

%12.4 MODELED Saturation transfer characteristic curve

for Vgs = 1:1:N
idc_Vd2(Vgs) = ((K*ufet0)/(1+(R*(K*ufet0)*(((Vgs-1)/(N-1))*VgsmaxS)-Vt)^(1+gamma))))*...
(((Vgs-1)/(N-1))*VgsmaxS)-Vt)^(1+gamma))*(Vd2)*(1+lambda*(Vd2)))...
((1+((Vd2)/(alphas*(((Vgs-1)/(N-1))*VgsmaxS)-Vt)))^m)^(1/m));
end

% Step 13: Calculation of real IDS currents by the addition of the
% correction factor CC: IDREAL = IDSMODELED + VDS/CC

Vgsz = length(Vgs_out);
Vdsze = Vdsz;
idcreal = zeros(Vgsz, (N));
idcreal_Vgs = zeros(Vgsz, (N));
idcreal_lm = zeros(Vgsz, Vdsze);
ireal_lm = zeros(Vgsz, Vdsze);
idcreal_Vd1 = zeros(1, N);
idc_Vd2cor = zeros(1, N);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% REAL IDS CURRENT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 13.1. Real output characteristic curves: linear & saturation %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 13.2. Real Output characteristic curves in the linear region %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 13.3 Real Linear transfer characteristic curve %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 13.4 Real Saturation transfer characteristic curve %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% 13.1 REAL output characteristic curve in linear and saturation regions

for Vgs = 1:Vgsz
for Vds = 1:1:N;
idcreal(Vgs, Vds) = (((K*ufet0)/(1+(R*(K*ufet0)*((Vgs_out(Vgs)))-Vt)^(1+gamma))))*...
((((Vgs_out(Vgs)))-Vt)^(1+gamma))*(((Vds-1)/(N-1))*Vdmax))*(1+lambda*(((Vds-1)/(N-1))*Vdmax)))...
/((1+((((Vds-1)/(N-1))*Vdmax))/(alphas*(((Vgs_out(Vgs)))-Vt)))^m)^(1/m)))+( (((Vds-1)/(N-1))*Vdmax))/CC);
% For this part, Vgs has been changed to Vgs_out(Vgs), since Vgs is just an
% index, not the value itself. Hence, all the biasing points are accounted
end
end

for Vgs = 1:Vgsz
for Vds = 1:1:N;
idcreal_Vgs(Vgs, Vds) = idcreal(Vgs, Vds);
end
end

% 13.2 REAL output characteristic curve in linear region

for Vgs = 1:Vgsz

```

```

for Vds = 1:Vdsze
idcreal_lm(Vgs, Vds) = ((K*ufet0)/(1+(R*(K*ufet0)*...
((Vgs_out(Vgs))-Vt)^(1+gamma))))*(((Vgs_out(Vgs))-Vt)^(1+gamma))*(((Vds-1)/(N-1))*Vdmax)*...
(1+lambda*(((Vds-1)/(N-1))*Vdmax)))/((1+(((Vds-1)/(N-1))*Vdmax)/(alphas*((Vgs_out(Vgs))-Vt)))^m)^(1/m))+...
(((Vds-1)/(N-1))*Vdmax)/CC);
end
end

wo
= w0;

% Interpolated measured data

for Vgs = 1:Vgsz
for Vds = 1:Vdsze
ireal_lm(Vgs, Vds) = ireal_inter(Vgs, Vds);
end
end

%13.3 REAL Linear transfer characteristic curve

for Vgs = 1:1:N
idcreal_Vd1(Vgs) = ((K*ufet0)/(1+(R*(K*ufet0)*...
(((Vgs-1)/(N-1))*Vgsmax)-Vt)^(1+gamma))))*((((Vgs-1)/(N-1))*Vgsmax)-Vt)^(1+gamma))*((Vd1)*(1+lambda*(Vd1)))...
/((1+((Vd1)/(alphas*(((Vgs-1)/(N-1))*Vgsmax)-Vt)))^m)^(1/m)))+(Vd1)/CC);
end

%13.4 REAL Saturation transfer characteristic curve

for Vgs = 1:1:N
idc_Vd2cor(Vgs) = ((K*ufet0)/(1+(R*(K*ufet0)*(((Vgs-1)/(N-1))*VgsmaxS)-Vt)^(1+gamma))))...
*((((Vgs-1)/(N-1))*VgsmaxS)-Vt)^(1+gamma))*((Vd2)*(1+lambda*(Vd2)))...
/((1+((Vd2)/(alphas*(((Vgs-1)/(N-1))*VgsmaxS)-Vt)))^m)^(1/m)))+(Vd2)/CC);
end

% Step 14. Calculation of the field mobility graphic
% This mobility block do the same as before. It can be neglected.

ufet_mob
= zeros(1,length(Vg_LTC));

for i = 1:length(idllkreal)
ufet_mob(i) = ufet0*((xx_spline(i)-Vt)^gamma);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 2. SUB THRESHOLD REGIME %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 2.1 Parameters to calculate the sub-threshold regime

% Extrapolation to find Ioff is just made when Vt is negative, in order to avoid significant estimation errors

```

```

if Vt < 0

MIN_LINEAR          = (-4/30)*(Vgsmax);
xxx_spline          = MIN_LINEAR:(Vgsmax)/(N-1):Vgsmax;
idllkreal1          = spline(Vg_LTC, Vg_LTC_idlreal,xxx_spline);

else
idllkreal1 = 0;
end

Io                  = idllkreal1(1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 3. ESTRADA'S COMPACT MODEL FOR CURRENT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% 3.1 Transfer characteristic curve Improvement of the Linear Modeling

ISc                  = zeros(1,N+5);
Itc                  = zeros(1,N+5);
Itc_plot             = zeros(1,N+5);
idc_Vds5_Vgs_expanded = zeros(1,N+5);

idc_Vd1_inter        = interp1(idc_Vd1_interpolant, idc_Vd1 , idc_Vd1_inter_ISc, 'linear');

% Modification made above in order to show how the number of Vt decimal
% places are considered to calculate the subthreshold current Is. Is is
% shown as ISc.

for Vgs = 1:1:(N+5)
idc_Vds5_Vgs_expanded(Vgs) = ((K*ufet0)/(1+(R*(K*ufet0)*(((Vgs-6)/(N-1))*Vgsmax)-Vt)^(1+gamma))))...
*(((Vgs-6)/(N-1))*Vgsmax)-Vt)^(1+gamma))*(Vd1)*(1+lambda*(Vd1)))...
/((1+((Vd1)/(alphas*(((Vgs-6)/(N-1))*Vgsmax)-Vt)))^m)^(1/m));
end

% Model adjusting parameters optimization below
% (the "max" or "mean" strategies can be switched in one of the lines shown below)

p                    = 0;
err                  = zeros(4,400000);
S                    = zeros(1,100);
DV                   = zeros(1, 40);
Q1                   = zeros(1, 100);

for j = 1:1:100
S(1,j)              = 0.2*(j-1);
for jj = 1:1:40
DV(1,jj)            = 0.1*(jj);
for jjj = 1:1:100
Q1(1,jjj)           = 0.1*(jjj-1);
for Vgs = 1:1:(N+5)
ISc(Vgs) = idc_Vd1_inter(ceil((Vt+DV(1,jj))*1000)+1)*(exp((((Vgs-6)/(N-1))*Vgsmax)-Vt)/S(1,j))*2.3);
end
for Vgs = 1:1:(N+5)

```



```

Itc(Vgs) = ((abs(ISc(Vgs)))*((1-(tanh((((Vgs-6)/(N-1))*Vgsmax)-(Vt+DV(1,jj))*Q1(1,jj)))/(2)))...
+((abs(idc_Vds5_Vgs_expanded(Vgs))*((1+(tanh((((Vgs-6)/(N-1))*Vgsmax)-(Vt+DV(1,jj))*Q1(1,jj)))/(2)))));
end
p = p+1; %%%%% Change between "max" or "mean" in the line below %%%%%
err(1, p) = mean(abs(idllkreal(2:1:end) - Itc(1,(7:1:end)))./idllkreal(2:1:end));
err(2,p) = S(1,j);
err(3,p) = DV(1,jj);
err(4,p) = Q1(1,jj);
end
end
end

min_err = min(err(1,:));
pos_min_err = find(abs(min_err - err(1,:)) < 1e-10);
S          = err(2,pos_min_err(1));
DV          = err(3,pos_min_err(1));
Q1          = err(4, pos_min_err(1));

SS          = S;
DVS         = DV;
Q1S         = Q1;

% Currents below are calculated again with the optimized paramters found above

for Vgs = 1:1:(N+5)
ISc(Vgs) = idc_Vd1_inter(ceil((Vt+DV)*1000)+1)*(exp((((Vgs-6)/(N-1))*Vgsmax)-(Vt)/S)*2.3));
end

for Vgs = 1:1:(N+5)
Itc(Vgs) = ((abs(ISc(Vgs)))*((1-(tanh((((Vgs-6)/(N-1))*Vgsmax)-(Vt+DV))*Q1)))/(2)))...
+((abs(idc_Vds5_Vgs_expanded(Vgs))*((1+(tanh((((Vgs-6)/(N-1))*Vgsmax)-(Vt+DV))*Q1)))/(2)))));
end

for Vgs = 1:1:(N+5)
Itc_plot(Vgs) = Itc(Vgs)+Io;
end

% 3.2 Transfer characteristic curve Improvement of the Saturation Modeling

idc_Vd2_Itc          = zeros(1,N+5);
IScS                 = zeros(1,N+5);
ItcS                 = zeros(1,N+5);
ItcS_plot            = zeros(1,N+5);

% In order to avoid data extrapolation, the fucntions are calculated again below
% for a higher range of voltages (including negative Vgs values)

for Vgs = 1:1:(N+5)
idc_Vd2_Itc(Vgs) = ((K*ufet0)/(1+(R*(K*ufet0)*((((Vgs-6)/(N-1))*VgsmaxS)-(Vt)^(1+gamma))))...
*((((Vgs-6)/(N-1))*VgsmaxS)-(Vt)^(1+gamma))*Vd2*(1+lambda*(Vd2)))...
/((1+((Vd2)/(alphas*(((Vgs-6)/(N-1))*VgsmaxS)-(Vt)))^m)^(1/m)));
end

idc_Vd2_inter          = interp1(idc_Vd2_interpolant, idc_Vd2, idc_Vd2_inter_ISc, 'linear');

for Vgs = 1:1:(N+5)

```

```
IScS(Vgs) = idc_Vd2_inter(ceil((Vt+DVS)*1000)+1)*(exp((((Vgs-6)/(N-1))*VgsmaxS-Vt)/SS)*2.3));
end
```

```
for Vgs = 1:1:(N+5)
ItcS(Vgs) = ((abs(IScS(Vgs)))*((1-(tanh((((Vgs-6)/(N-1))*VgsmaxS)-(Vt+DVS))*Q1S))/(2)))...
+((abs(idc_Vd2_Itc(Vgs)))*((1+(tanh((((Vgs-6)/(N-1))*VgsmaxS)-(Vt+DVS))*Q1S))/(2)));
end
```

```
for Vgs = 1:1:(N+5)
ItcS_plot(Vgs) = ItcS(Vgs)+IoS;
end
```

% 3.3 Output characteristic curves result

```
ISc = zeros(length(Vgs_out), N);
Itc_Vgs = zeros(length(Vgs_out), N);
idc_Itc = zeros(1, N);
Itc_Vgs_plot = zeros(length(Vgs_out), N);
```

```
Vgs_Vt = Vt + DV;
for Vds = 1:1:N
idc_Itc(Vds) = ((K*ufet0)/(1+(R*(K*ufet0)*((Vgs_Vt)-Vt)^(1+gamma))))...
*(((Vgs_Vt)-Vt)^(1+gamma))*(((Vds-1)/(N-1))*Vdmax))*(1+lambda*(((Vds-1)/(N-1))*Vdmax)))...
/((1+(((Vds-1)/(N-1))*Vdmax))/(alpha*(Vgs_Vt)-Vt))^m)^(1/m));
end
```

```
for Vgs = 1:length(Vgs_out)
for Vds = 1:1:N
ISc(Vgs, Vds) = idc_Itc(Vds)*(exp(((Vgs_out(Vgs)-Vt)/S)*2.3));
end
end
```

```
for Vgs = 1:length(Vgs_out)
for Vds = 1:1:N
Itc_Vgs(Vgs, Vds) = ((abs(ISc(Vgs,Vds)))*((1-(tanh((Vgs_out(Vgs)-(Vt+DV))*Q1)))/(2)))...
+((abs(idc_Vgs(Vgs,Vds)))*((1+(tanh((Vgs_out(Vgs)-(Vt+DV))*Q1)))/(2)));
end
end
```

```
for Vgs = 1:length(Vgs_out)
for Vds = 1:1:N
Itc_Vgs_plot(Vgs, Vds) = Itc_Vgs(Vgs, Vds)+(((Vds-1)/(N-1))*Vdmax)/CC;
end
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 4. DENSITY OF STATES %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

% This block hasn't been used so far. Maybe it becomes important with the addition
% of non-ohmics contacts.

```
T2 = (1/P_Vt)*(T/2);
ffc = (1/Vaa)^((2*T2/T)-2);
```

```

Ntef = (Ci/q)*(((S)/(2.3*Vth))-1);

% 4.1 Calculation of gdo
es = esr*e0;
FF1 = (Ci^((2*T2/T)-2))/((es^((T2/T)-1))*((2*kb*T2)^(T2/T)));
FF2 = q*Nv*(kb*T)*(((sin((pi)*(T/T2)))/((pi)*(kb*T)*q))^(T2/T));
gdo = ((FF1*FF2)/(ffc))^(T/T2);
Ea = kb*T2;
Energy = 0:0.005:0.5;
DOS = gdo*exp(-(Energy/Ea));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 5.Transconductance calculation %%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% New part of this code

% For linear transfer function

% Experimental

Vgs_Vd1_interp = 0:0.001:Vgsmax;
idllkreal_interp = spline(Vgs_Vd1, idllkreal, Vgs_Vd1_interp);
gm_lin_exp = diff(idllkreal_interp,1);
Vgs_lin1_diff = diff(Vgs_Vd1_interp, 1);
gm_lin_exp = gm_lin_exp./Vgs_lin1_diff;
gm_lin_exp = spline(Vgs_Vd1_interp(1:1:end-1), gm_lin_exp, Vgs_Vd1(1:1:end-1));

% Modelled

xs1_interp = 0:0.001:Vgsmax;
Itc_plot_interp = spline(xs1(6:1:end), Itc_plot(6:1:end), xs1_interp);
gm_lin = diff(Itc_plot_interp, 1);
Vgs_lin2_diff = diff(xs1_interp, 1);
gm_lin = gm_lin./Vgs_lin2_diff;
gm_lin = spline(xs1_interp(1:1:end-1), gm_lin, xs1(6:1:end-1));

% For saturation transfer function

% Experimental

Vgs_Vd2_interp = 0:0.001:VgsmaxS;
iksatr_interp = spline(Vgs_Vd2, iksatr, Vgs_Vd2_interp);
gm_sat_exp = diff(iksatr_interp,1);
Vgs_sat1_diff = diff(Vgs_Vd2_interp, 1);
gm_sat_exp = gm_sat_exp./Vgs_sat1_diff;
gm_sat_exp = spline(Vgs_Vd2_interp(1:1:end-1), gm_sat_exp, Vgs_Vd2(1:1:end-1));

% Modelled

xs3_interp = 0:0.001:VgsmaxS;
ItcS_plot_interp = spline(xs3(6:1:end), ItcS_plot(6:1:end), xs3_interp);
gm_sat = diff(ItcS_plot_interp, 1);
Vgs_sat2_diff = diff(xs3_interp, 1);
gm_sat = gm_sat./Vgs_sat2_diff;
gm_sat = spline(xs3_interp(1:1:end-1), gm_sat, xs3(6:1:end-1));

```

```

% Plots

figure(6)

plot(xs3(6:1:end-1), gm_sat, '-k', 'LineWidth', 5);
hold on
plot(Vgs_Vd2(1:1:end-1), gm_sat_exp, 'ob', 'MarkerFaceColor','b', 'MarkerSize', 7);
legend('Transcondutância(MATLAB) STC' , 'Transcondutância experimental STC');
if Stype > 1
ylabel('G_m (S)', 'FontSize',16, 'Fontweight', 'b');
xlabel('- V_g_s (V)', 'FontSize',16, 'Fontweight', 'b');
else
ylabel('G_m (S)', 'FontSize',16, 'Fontweight', 'b');
xlabel('V_g_s (V)', 'FontSize',16, 'Fontweight', 'b');
end

set(gca, 'FontSize',16, 'Fontweight', 'b');
hold off

figure(7)

plot(xs1(6:1:end-1), gm_lin, '-k', 'LineWidth', 5);
hold on;
plot(Vgs_Vd1(1:1:end-1), gm_lin_exp, 'ob', 'MarkerFaceColor','b', 'MarkerSize', 7);
legend('Transcondutância(MATLAB) LTC', 'Transcondutância experimental LTC');
if Stype > 1
ylabel('G_m (S)', 'FontSize',16, 'Fontweight', 'b');
xlabel('- V_g_s (V)', 'FontSize',16, 'Fontweight', 'b');
else
ylabel('G_m (S)', 'FontSize',16, 'Fontweight', 'b');
xlabel('V_g_s (V)', 'FontSize',16, 'Fontweight', 'b');
end

set(gca, 'FontSize',16, 'Fontweight', 'b');
hold off;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 6.Graphic Simulation %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% New part of the code

% 1 - Output characteristics for overall below and above threshold regions

figure(1)

plot(w1, Itc_Vgs_plot(1,:), '-k', 'LineWidth', 5);
hold on;
plot(w1,ireal(1,:), 'ob', 'MarkerFaceColor','b', 'MarkerSize', 7);
for iii = 2:length(Itc_Vgs_plot(:,1))
plot(w1, Itc_Vgs_plot(iii,:), '-k', 'LineWidth', 5);
plot(w1,ireal(iii,:), 'ob', 'MarkerFaceColor','b', 'MarkerSize', 7);
end

```

```

legend('UMEM(MATLAB)', 'Experimental');

if Stype > 1
ylabel('- I_d_s (A)', 'FontSize',16, 'Fontweight', 'b');
xlabel('- V_d_s (V)', 'FontSize',16, 'Fontweight', 'b');
else
ylabel('I_d_s (A)', 'FontSize',16, 'Fontweight', 'b');
xlabel('V_d_s (V)', 'FontSize',16, 'Fontweight', 'b');
end

set(gca, 'FontSize',16, 'Fontweight', 'b');

hold off;

% 2- STC for below and above threshold regions

figure(2)

plot(xs3,ItcS_plot, '-k', 'LineWidth', 5);
hold on;
plot(Vgs_Vd2, iksatr, 'ob', 'MarkerFaceColor','b', 'MarkerSize', 7);
legend('UMEM(MATLAB) STC', 'Experimental STC');
if Stype > 1
ylabel('- I_d_s (A)', 'FontSize',16, 'Fontweight', 'b');
xlabel('- V_g_s (V)', 'FontSize',16, 'Fontweight', 'b');
else
ylabel('I_d_s (A)', 'FontSize',16, 'Fontweight', 'b');
xlabel('V_g_s (V)', 'FontSize',16, 'Fontweight', 'b');
end
set(gca, 'FontSize',16, 'Fontweight', 'b');
hold off;

% 1 - LTC for below and above theshold regions

figure(3)

plot(xs1,Itc_plot, '-k', 'LineWidth', 5);
hold on;
plot(Vgs_Vd1, idllkreal, 'ob', 'MarkerFaceColor','b', 'MarkerSize', 7);
legend('UMEM(MATLAB) LTC', 'Experimental LTC');
if Stype > 1
ylabel('- I_d_s (A)', 'FontSize',16, 'Fontweight', 'b');
xlabel('- V_g_s (V)', 'FontSize',16, 'Fontweight', 'b');
else
ylabel('I_d_s (A)', 'FontSize',16, 'Fontweight', 'b');
xlabel('V_g_s (V)', 'FontSize',16, 'Fontweight', 'b');
end
set(gca, 'FontSize',16, 'Fontweight', 'b');
hold off;

```

```
% EXTRACTED, MODEL ADJUSTING AND EXTRACTION ADJUSTMENT PARAMETERS PLUS  
% NUMBER OF POINTS AND PHYSICAL TFT PARAMETERS
```

```
gamma  
Vaa  
R  
alphas  
m  
lambda  
Vt  
eir  
W  
L  
di  
Io  
DV  
S  
Q1  
IoS  
DVS  
SS  
Q1S  
CC  
Vgs_Hmin  
Vgs_Hmax  
Vgs_ips1_min  
Vgs_ips1_max  
N
```

II. CÓDIGO EM VERILOG A

```
//MODEL FOR OTFT BELOW AND ABOVE THRESHOLD
// CINVESTAV-IPN 2012

`include "constants.vams"
`include "discipline.vams"
`define P_K (1.3806226e-23) //Boltzmann cionstant in eV/K
`define P_Q (1.6021918e-19) //Charge Cb
`define M_PI (3.14159265358979323846)
`define P_E0 (8.854214871e-14) //VACCUM PERMITTIVITY IN [F/cm]
// UNITS CONVERTED FROM F/M TO F/CM ABOVE

//Define module with 3 terminals
module otftmi(d, g, s);

//Declare input/output nodes ("inout")
inout d, g, s;

// type of nodes "electrical"
electrical d, g, s;

// Internal nodes
branch (d, s) canal;

//Parameter definition

// TFT geometry
parameter real TYPE=-1 from[-1:1] exclude 0; //1.0 N-TYPE, -1.0 P-TYPE
parameter real L = 50e-4 from[0.0:inf]; //CHANNEL LENGTH IN [cm]
parameter real W = 1000e-4 from[0.0:inf]; //CHANNEL WIDTH IN [cm]

// DEFAULT PARAMETERS TO BE CHANGED EXTERNALLY

//DEFAULT MODEL PARAMETERS
parameter real ALPHASAT=0.572; //SATURATION PARAMETER IN [eV]
parameter real EPS=9.4; // RELATIVE DIELECTRIC CONSTANT OF THE SEMICONDUCTOR
parameter real EPSI=2.6; //RELATIVE DIELECTRIC CONSTANT OF THE GATE DIELECTRIC
parameter real GAMMA=1.08; // MOBILITY DEPENDENCE ON (VGS-VT)
parameter real LAMBDA=-1.3e-3; //OUTPUT CONDUCTANCE IN [1/V]
parameter real M=1.475; //INFLECTION PARAMETER OF OUTPUT CHARACTERISTIC (KNEE)
parameter real RD = 0.474e6; //SERIES RESISTANCE AT DRAIN IN [Ohms]
parameter real RS = 0.474e6; //SERIES RESITANCE AT SOURCE IN [Ohms]
parameter real TNOM=27; //MEASUREMENT TEMPERATURE IN [C]
parameter real TOX=30e-7; //DIELECTRIC LAYER THICKNESS IN [cm]
parameter real V0=0.04; //CHARACTERISTIC VOLTAGE OF DEEP STATES IN [V]
parameter real VAA=0.634e1; //LOW FIELD MOBILITY PARAMETER
parameter real VTO=2.011; //THRESHOLD VOLTAGE IN [V]
parameter real SL=3; //SUBTRHESHOLD SLOPE OF LINEAR TRANSFER CURVE
parameter real DVL=2; //DEFINES VOLTAGE FOR SEWING BELOW-ABOVE THRESHOLD REGIONS OF LINEAR TRANSFER CURVE IN [V]
parameter real QL=0.4; //TANH PARAMETER SEWING BELOW-ABOVE THRESHOLD REGIONS OF LINEAR TRANSFER CURVE
parameter real IOL=0; //IOFF FOR LINEAR TRANSFER CURVE IN [A]IOL=1.006e-9; // NEGLECTED
parameter real SS=3; //SUBTRHESHOLD SLOPE OF TRANSFER CURVE IN SATURATION
parameter real DVS=2; //DEFINES VOLTAGE FOR SEWING BELOW-ABOVE THRESHOLD REGIONS OF SATURATION TRANSFER CURVE IN [V]
parameter real QS=0.4; //TANH PARAMETER SEWING BELOW-ABOVE THRESHOLD REGIONS OF SATURATION TRANSFER CURVE
parameter real IOS=0; //IOFF FOR SATURATION TRANSFER CURVE IN [A]IOS=5.144e-9; // NEGLECTED
```

```

//ALL TRANSISTOR DIMENSIONS, W, L AND TOX WERE CONVERTED FROM METERS TO CENTIMETERS

// DEFAULT MODEL PARAMETERS FOR NON-OHMIC CONTACT AT D AND S

parameter real ETAD=2.75; //RESISTANCE DRAIN VOTAGE DEPENDENCE ADJUSTING PARAMETER
parameter real ETAG=2.75; //RESISTANCE GATE VOLTAGE DEPENDENCE ADJUSTING PARAMETER
parameter real RCO=1e13; //ADJUSTING PARAMETER

//DEFAULT MODEL PARAMETERS FOR SMICONDUCTOR LAYER CONDUCTANCE BETWEEN D AND S

parameter real CCt=3.5e9; //SEMICONDUCTOR LAYER CONDUCTANCE IN [S]

//INTERNAL VARIABLES
real EPSIO, EPSO;

//OTHER VARIABLES
real MOS25cg, MUFET,CC;

//TEMPERATURE DEPENDENT VARIABLES
real TNOMO;
real VT;

//RESISTANCE RELATED VARIABLES
real R;
real RO;
real RC;
real RT;

// VOLTAGE RELATED VARIABLES
real Vds, Vgs, Vgd;
real vgs, vgd, vds, mode;
real vgt, vgte, vdse, vsate;

// VARIABLES IN BELOW THRESHOLD
real vgtgL, vgtedL, vsatedL, vdsedL;
real nsadL, gchidL, gchdL, idcdL;
real idscL;

// VARIABLES IN ABOVE THRESHOLD
real vgtgS, vgtedS, vsatedS, vdsedS;
real nsadS, gchidS, gchdS, idcdS;
real idscS;
//VARIABLES FOR COMPLETE BELOW AND ABOVE REGIONS
real idc, ids, idsL, idsS, ifinal;
// OTHER VARIABLES REQUIRED
real nsa, gchi, gch, VMIN, DELTA;

analog
begin
//INITIAL CALCULATIONS
RO = 1;
VMIN = 0.01;
DELTA = 1;
TNOMO =TNOM+'P_CELSIUSO;
EPSIO = 'P_EO * EPSI;
EPSO = 'P_EO * EPS;
VT= (TYPE)*VT0;
// VT0 is not the extracted, but Vt = S*VT from MATLAB. This operation just convert it back into a positive value
MOS25cg = EPSIO / TOX;

```



```

MUFET = 1/pow(VAA,GAMMA);
CC=CCt; // CCt/(W/L);
// CCt is considered equal to the correction factor extracted...
//in MATLAB in order to account for hysteresis effects

Vds = TYPE * V(d, s);
Vgs = TYPE * V(g, s);
Vgd = TYPE * V(g, d);
//NODE INITIALIZATION
if (Vds >= 0.0)
begin
mode = 1;
vds = Vds;
vgs = Vgs;
vgd = Vgd;
end
else
begin //modo inverso
mode = -1;
vds = -Vds;
vgs = Vgd;
vgd = Vgs;
end

//CALCULATION OF OHMIC AND NON-OMIC RESISTANCE AT CONTACTS
R = RS + RD;
RC = R0 + (RC0 * exp(-ETAD * vds) * exp(-ETAG * vgs));
RT = R + RC;
// EXPRESSION TO CALCULATE CURVES AT VG=VT
vgt = (VMIN/2) * (1+((vgs-VT)/VMIN) + sqrt((DELTA*DELTA)+pow((((vgs-VT)/VMIN)-1),2)));
//CALCULATION OF BELOW THRESHOLD LINEAR CHARACTERISTICS
vgtdL = VT + DVL;
vgtdL = vgtdL - VT;
vsatedL = ALPHASAT * vgtdL;
vdsedL = vds / (pow((1 + (pow((vds/vsatedL),M))), (1 / M)));
nsadL = (pow(vgtdL ,GAMMA)) * vgtdL * (MOS25cg / 'P_Q');
gchidL = 'P_Q * nsadL * MUFET * (W / L);
gchdL = gchidL / (1 + (gchidL * RT));
idcdL = gchdL * vdsedL * (1 + (LAMBDA * vds));
idscL = idcdL * exp(((vgs - VT) / SL) * 2.3);
//CALCULATION OF BELOW THRESHOLD CHARACTERISTICS IN SATURATION
vgtdS = VT + DVS;
vgtdS = vgtdS - VT;
vsatedS = ALPHASAT * vgtdS;
vdsedS = vds / (pow((1 + (pow((vds/vsatedS),M))), (1 / M)));
nsadS = (pow(vgtdS ,GAMMA)) * vgtdS * (MOS25cg / 'P_Q');
gchidS = 'P_Q * nsadS * MUFET * (W / L);
gchdS = gchidS / (1 + (gchidS * RT));
idcdS = gchdS * vdsedS * (1 + (LAMBDA * vds));
idscS = idcdS * exp(((vgs-VT) / SS) * 2.3);
//CALCULATION OF ABOVE THRESHOLD CHARACTERISTICS
vsate = ALPHASAT * vgt;
vdse = vds / (pow((1 + (pow((vds/vsate),M))), (1 / M)));
nsa = (pow(vgt ,GAMMA)) * vgt * (MOS25cg / 'P_Q');
gchi = 'P_Q * nsa * MUFET * (W / L);
gch = gchi / (1 + (gchi * RT));
idc = (gch * vdse * (1 + (LAMBDA * vds))) ;
//CALCULATION OF CURRENTS
idsL = abs(idscL) * ((1 - tanh((vgs - vgtdL) * QL)) / 2) + abs(idc) * ((1 + tanh((vgs - vgtdL) * QL)) / 2) + IOL + (vds/CC);
idsS = abs(idscS) * ((1 - tanh((vgs - vgtdS) * QS)) / 2) + abs(idc) * ((1 + tanh((vgs - vgtdS) * QS)) / 2) + IOS + (vds/CC);

```

```

if (vgs < VT0)          // Correction factor added to equations above. IO is considered to be equal to zero.
ids = idsL;
else
ids = idsS;
ifinal = ids;
if (mode > 0)
// I(canal) <+ TYPE * ifinal;
I(d, s) <+ TYPE * ifinal;
else
// I(canal) <+ -TYPE * ifinal;
I(d, s) <+ -TYPE * ifinal;
end

endmodule

```

III. ANEXOS DA SEÇÃO 4.3

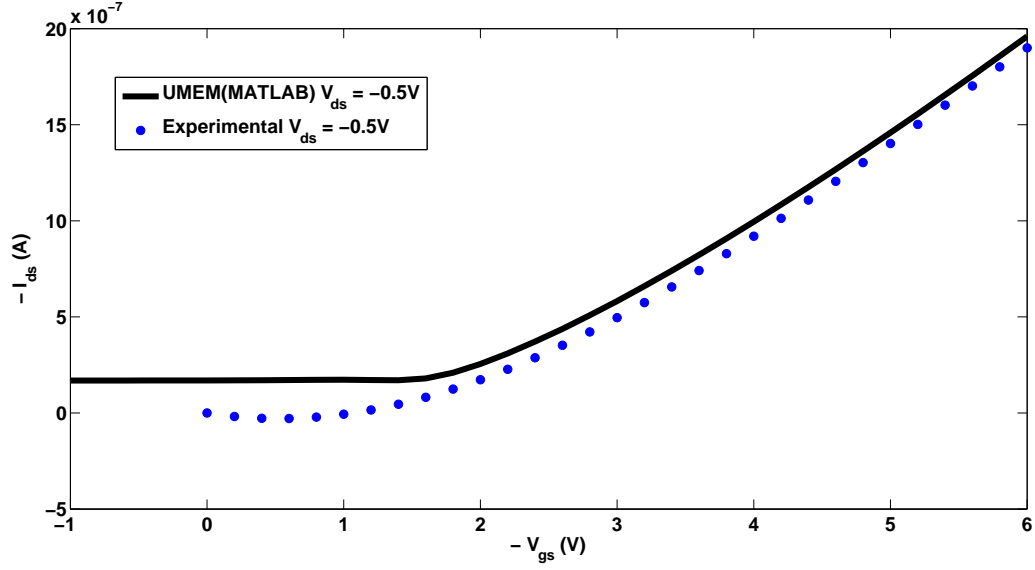


Figura III.1: Curvas características de transferência para o TUD OTFT encontradas experimentalmente e modelada pelo MATLAB para primeira simulação da tabela 4.2. $V_{H,min} = -4V$, $V_{H,max} = -6V$, $V_{IPC,min} = -4V$, $V_{IPC,max} = -6V$.

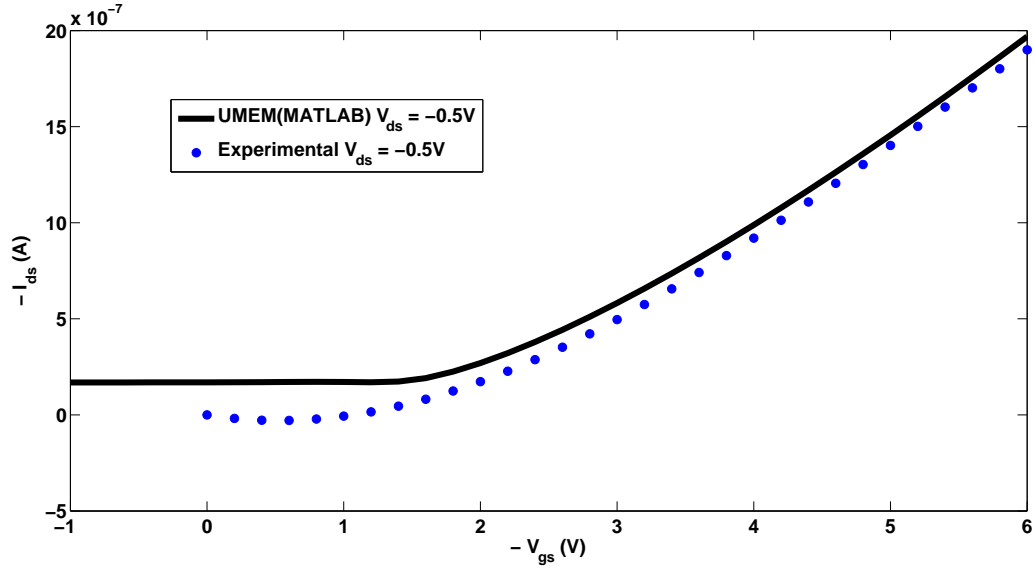


Figura III.2: Curvas características de transferência para o TUD OTFT encontradas experimentalmente e modelada pelo MATLAB para segunda simulação da tabela 4.2. $V_{H,min} = -3V$, $V_{H,max} = -5V$, $V_{IPC,min} = -4V$, $V_{IPC,max} = -6V$.

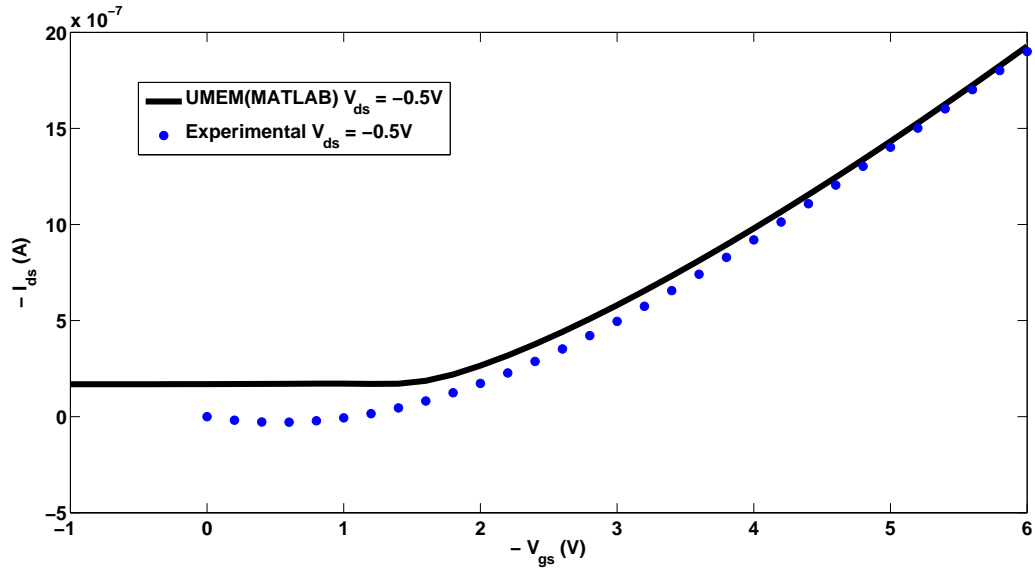


Figura III.3: Curvas características de transferência para o TUD OTFT encontradas experimentalmente e modelada pelo MATLAB para terceira simulação da tabela 4.2. $V_{H,min} = -2V$, $V_{H,max} = -4V$, $V_{IPC,min} = -4V$, $V_{IPC,max} = -6V$.

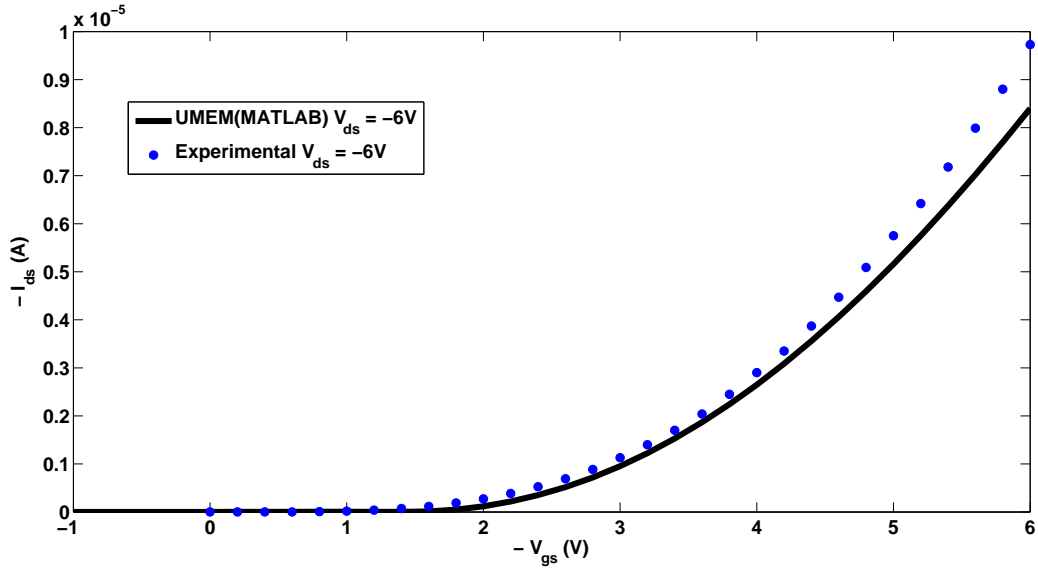


Figura III.4: Curvas características de transferência para o TUD OTFT encontradas experimentalmente e modelada pelo MATLAB para primeira simulação da tabela 4.2. $V_{H,min} = -4V$, $V_{H,max} = -6V$, $V_{IPC,min} = -4V$, $V_{IPC,max} = -6V$.

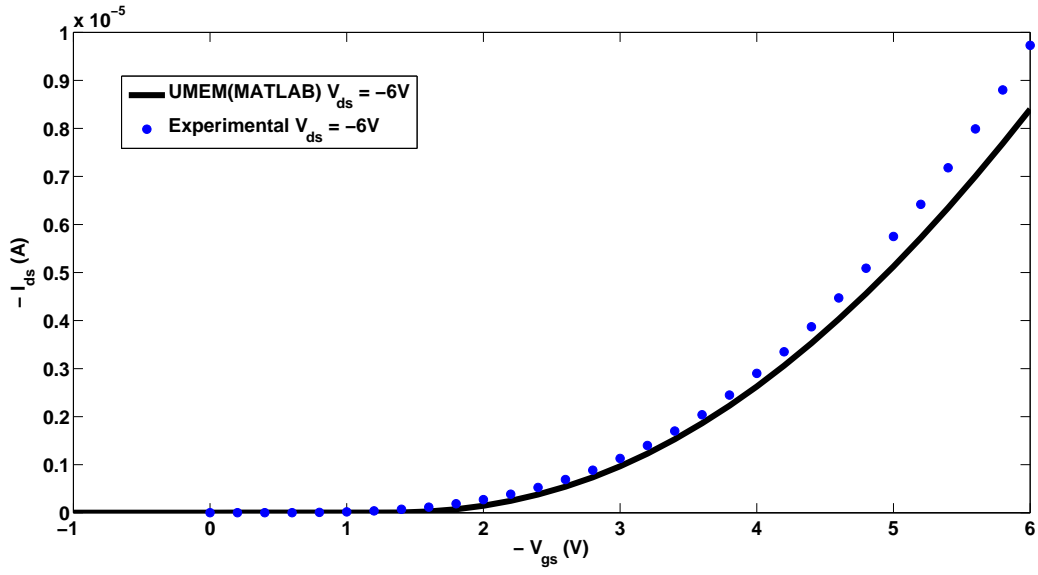


Figura III.5: Curvas características de transferência para o TUD OTFT encontradas experimentalmente e modelada pelo MATLAB para segunda simulação da tabela 4.2. $V_{H,min} = -3V$, $V_{H,max} = -5V$, $V_{IPC,min} = -4V$, $V_{IPC,max} = -6V$.

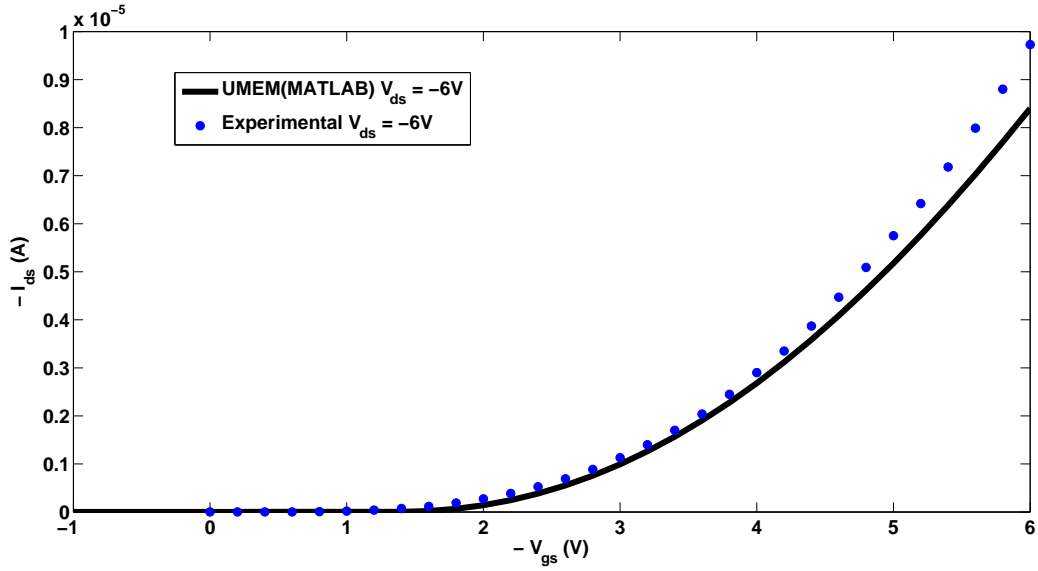


Figura III.6: Curvas características de transferência para o TUD OTFT encontradas experimentalmente e modelada pelo MATLAB para terceira simulação da tabela 4.2. $V_{H,min} = -2V$, $V_{H,max} = -4V$, $V_{IPC,min} = -4V$, $V_{IPC,max} = -6V$.

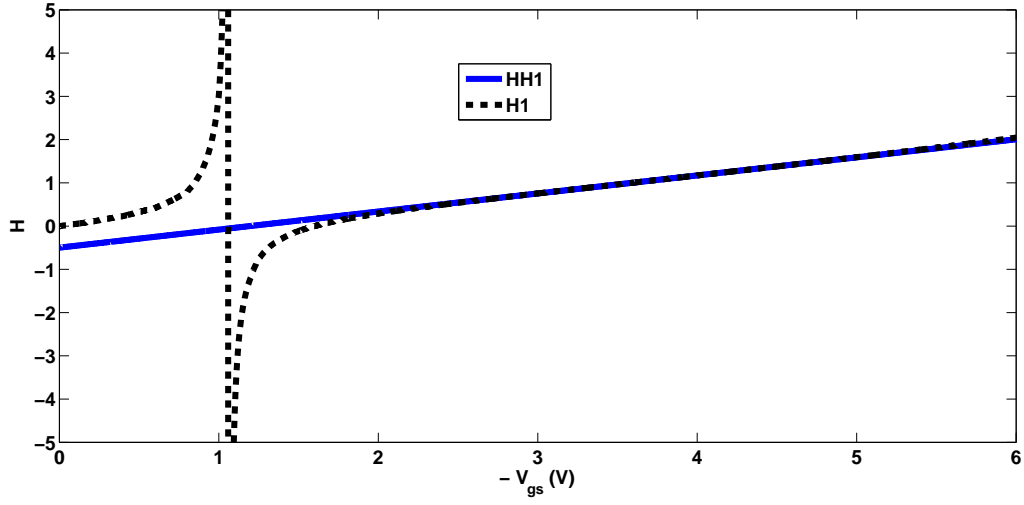


Figura III.7: Curvas das funções de ajuste $H(V_{gs})$ e $HH1(V_{gs})$ para o TUD OTFT usando os parâmetros da segunda simulação na tabela 4.2. $V_{H,min} = -3V$, $V_{H,max} = -5V$, $V_{IPC,min} = -4V$, $V_{IPC,max} = -6V$.

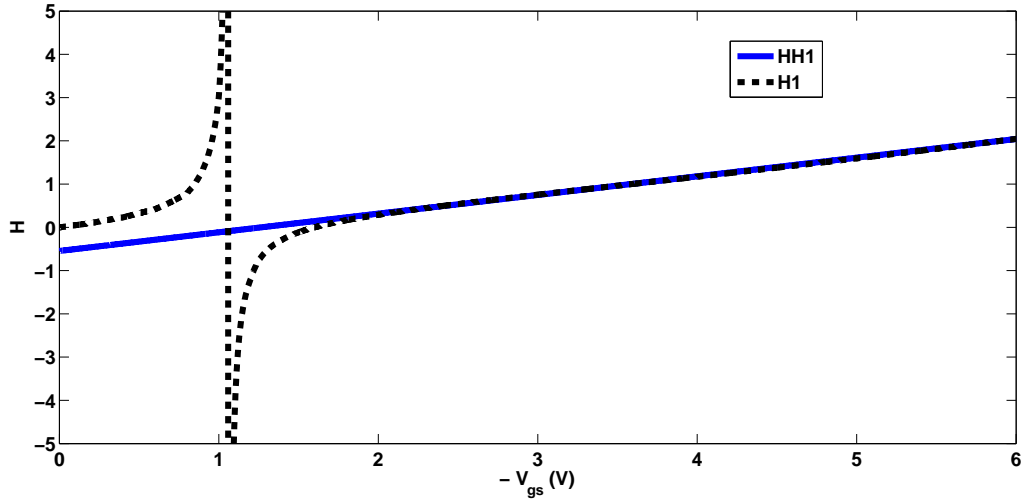


Figura III.8: Curvas das funções de ajuste $H(V_{gs})$ e $HH1(V_{gs})$ para o TUD OTFT usando os parâmetros da terceira simulação na tabela 4.2. $V_{H,min} = -2V$, $V_{H,max} = -4V$, $V_{IPC,min} = -4V$, $V_{IPC,max} = -6V$.

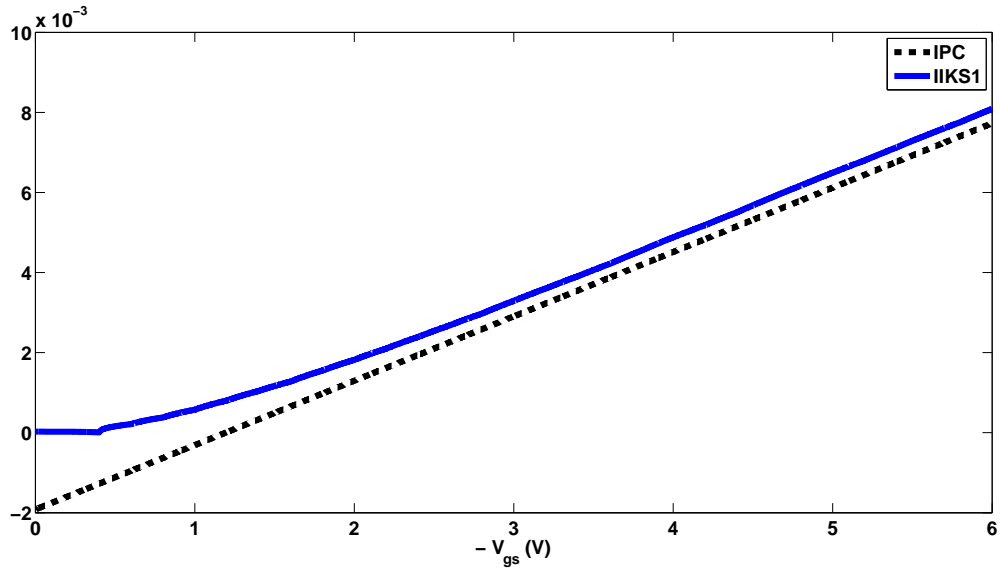


Figura III.9: Curvas das funções de ajuste $IPC(V_{gs})$ e $IKS1(V_{gs})$ para o TUD OTFT usando os parâmetros da segunda simulação na tabela 4.2. $V_{H,min} = -3V$, $V_{H,max} = -5V$, $V_{IPC,min} = -4V$, $V_{IPC,max} = -6V$.

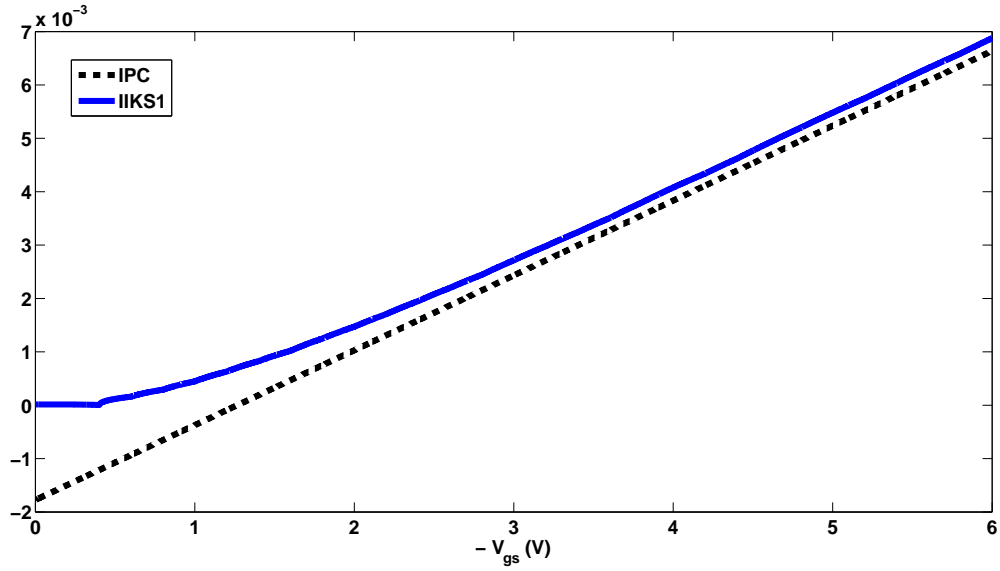


Figura III.10: Curvas das funções de ajuste $IPC(V_{gs})$ e $IKS1(V_{gs})$ para o TUD OTFT usando os parâmetros da terceira simulação na tabela 4.2. $V_{H,min} = -2V$, $V_{H,max} = -4V$, $V_{IPC,min} = -4V$, $V_{IPC,max} = -6V$.

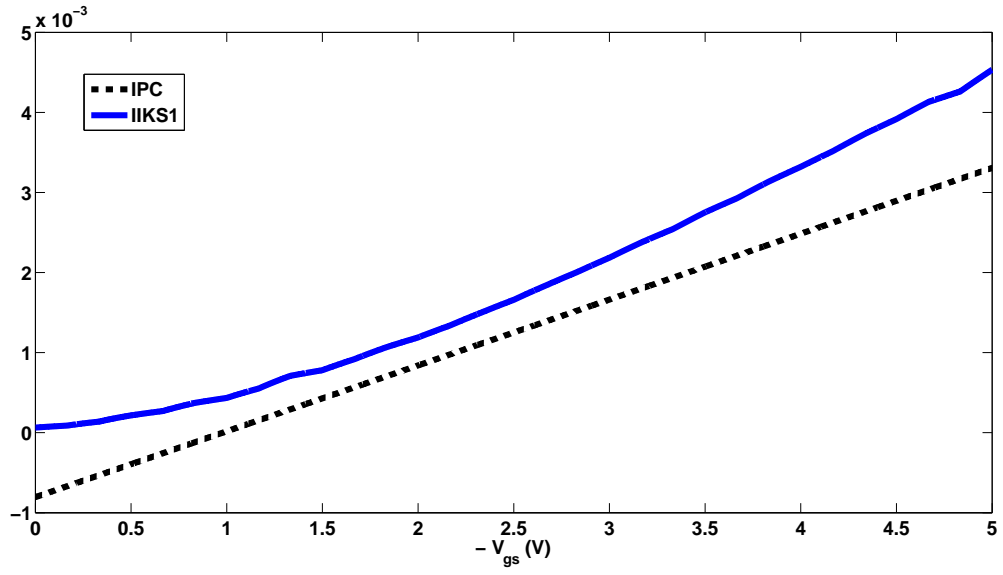


Figura III.11: Curvas das funções de ajuste $IPC(V_{gs})$ e $IKS1(V_{gs})$ para o TUD VOTFT usando os parâmetros da segunda simulação na tabela 4.4. $V_{H,min} = -4V$, $V_{H,max} = -6V$, $V_{IPC,min} = -1.8V$, $V_{IPC,max} = -2.2V$.

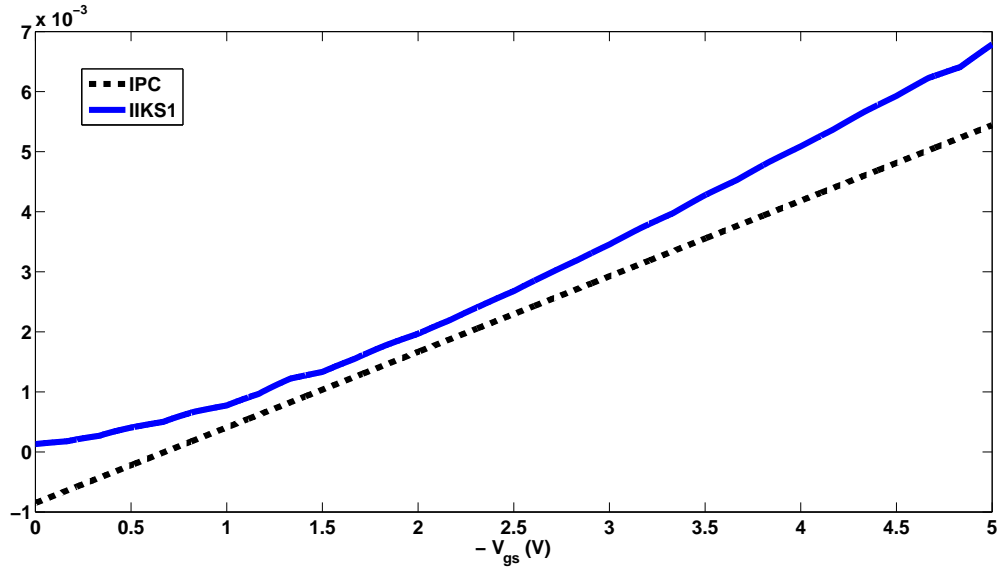


Figura III.12: Curvas das funções de ajuste $IPC(V_{gs})$ e $IKS1(V_{gs})$ para o TUD VOTFT usando os parâmetros da terceira simulação na tabela 4.4. $V_{H,min} = -3.5V$, $V_{H,max} = -5.5V$, $V_{IPC,min} = -1.8V$, $V_{IPC,max} = -2.2V$.

IV. ANEXOS DA SEÇÃO 4.4

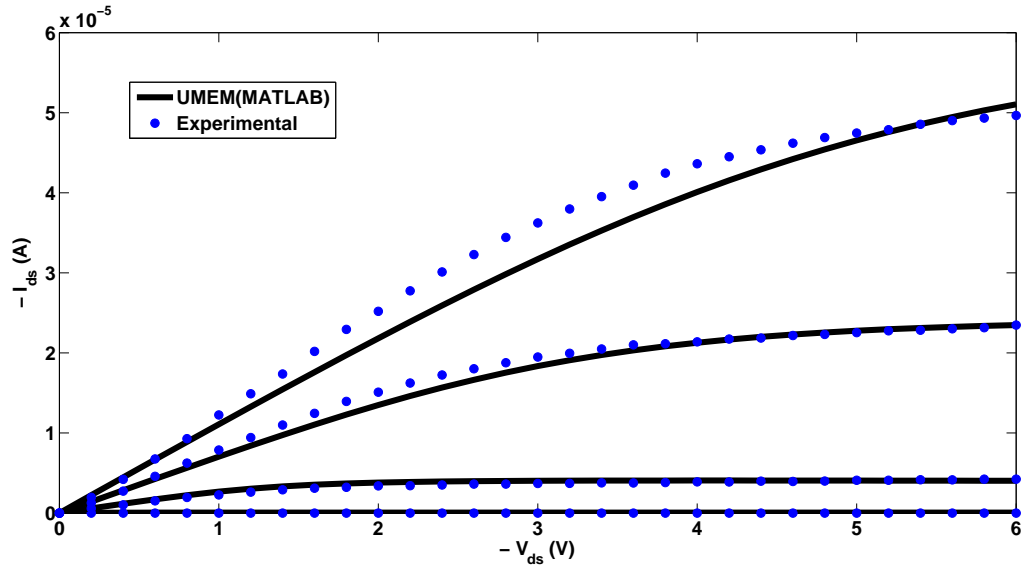


Figura IV.1: Curvas características de saída para o TUD VOTFT encontradas experimentalmente e modelada pelo MATLAB com $V_{gs} = 0V, -2V, -4V$ e $-6V$ através do método da menor média usando os parâmetros da tabela 4.6.

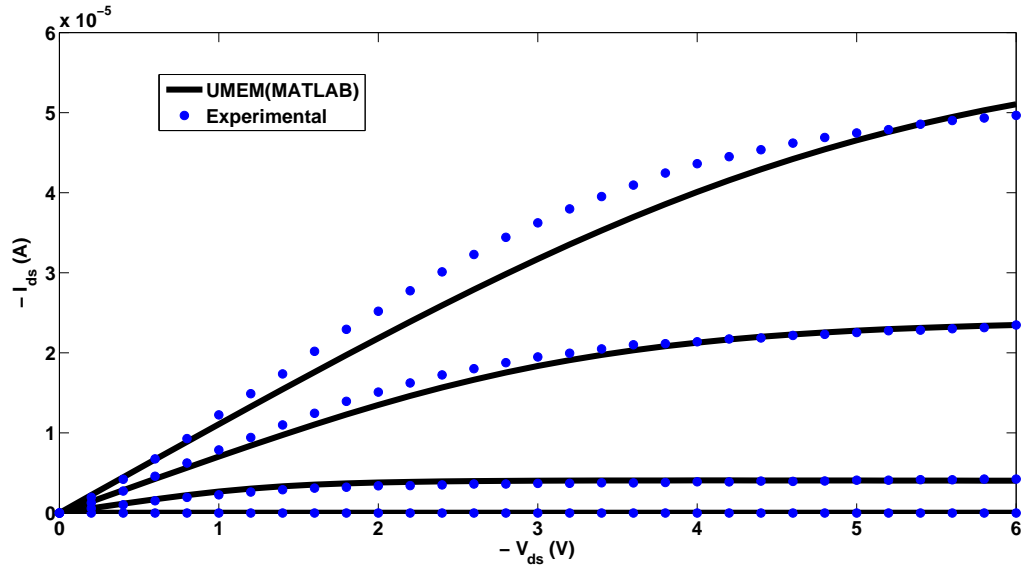


Figura IV.2: Curvas características de saída para o TUD VOTFT encontradas experimentalmente e modelada pelo MATLAB com $V_{gs} = 0V, -2V, -4V$ e $-6V$ através do método do menor valor máximo usando os parâmetros da tabela 4.6.